5 Remote Control – Basic Information

This chapter provides basic information on remote control, for example on the IEC/IEEE bus, RS-232-C interface, interface and device messages, command processing, status reporting system, etc.

The instrument is equipped with an IEC/IEEE-bus interface according to standard IEC 625.1/IEEE 488.1 and a RS-232-C interface. The connectors are located at the rear of the instrument and permit to connect a controller for remote control. The instrument supports the SCPI version 1994.0 (Standard Commands for Programmable Instruments). The SCPI standard is based on standard IEEE 488.2 and aims at the standardization of device-specific commands, error handling and the status registers.

For this section it is assumed that the user has basic knowledge of IEC/IEEE-bus programming and operation of the controller. A description of the interface commands will be found in the relevant manuals.
The requirements of the SCPI standard regarding command syntax, error handling and configuration of the status registers are explained in detail in the respective sections. Tables provide a fast overview of the bit assignment of the status registers. The tables are complemented by a comprehensive description of the status registers.
A description of commands is given in chapter 6. Programming examples for the main functions will be found in chapter 7.

Brief Instructions

The short and simple operating sequence given below permits fast putting into operation of the instrument and setting of its basic functions.

IEC/IEEE Bus

It is assumed that the IEC/IEEE-bus address, which is factory-set to 28, has not been changed.

1. Connect the instrument and the controller using the IEC/IEEE-bus cable.
2. Write and start the following program on the controller:

```
CALL IBFIND("DEV1", generator%)            Open port to instrument
CALL IBPAD(generator%, 28)                 Transfer instrument address to controller
CALL IBWRT(generator%, "*RST;*CLS")        Reset instrument
CALL IBWRT(generator%, "FREQ 1GHz")        Set frequency to 1 GHz
CALL IBWRT(generator%, "POW -7.3dBm")      Set output level to –7.3 dBm
CALL IBWRT(generator%, "OUTP:STAT ON")     Switch RF output on
CALL IBWRT(generator%, "AM:SOUR INT")      Set AM modulation source Lfgen
CALL IBWRT(generator%, "AM:INT:FREQ 15kHz")Set AM modulation frequency to 15 kHz
CALL IBWRT(generator%, "AM 30PCT")         Set AM modulation depth to 30%
CALL IBWRT(generator%, "AM:STAT ON")       Switch on AM
```

   An amplitude-modulated signal is now present at the output of the instrument.
3. To return to manual control, press the [LOCAL] key on the front panel.

RS-232-C Interface

It is assumed that the configuration of the RS-232-C interface of the unit has not yet been changed.

1. Connect the unit and the controller using the null modem cable.
2. Enter the following command on the controller to configure the controller interface:
   mode com1: 9600, n, 8, 1
3. Create the following ASCII file on the controller:

```
                                  Switch instrument to remote control (RETURN)
*RST;*CLS                         Reset instrument
FREQ 1GHz                         Set frequency to 1 GHz
POW -7.3dBm                       Set output level to –7.3 dBm
OUTP:STAT ON                      Switch on RF output
AM 30PCT                          Set AM modulation depth to 30%
AM:STAT ON                        Switch on AM
                                  (RETURN)
```

4. Transfer the ASCII file to the instrument via the RS-232-C interface. Enter the following command on the controller:
   copy <filename> com1:
   An amplitude-modulated signal is now present at the output of the instrument.
5. To return to manual control, press the [LOCAL] key on the front panel.

Switchover to Remote Control

On power-up, the instrument is always in the manual control mode ("LOCAL" state) and can be operated via the front panel.

The instrument is switched to remote control ("REMOTE" state) as follows:

IEC/IEEE-bus: when it receives an addressed command from the controller.

RS-232-C interface: when it receives a carriage return <CR> (=0Dh) or a line feed <LF> (=0Ah) from the controller.

During remote control, operation via the front panel is disabled. The instrument remains in the remote state until it is reset to the manual state via the front panel or via the IEC/IEEE bus. Switching from manual to remote control and vice versa does not affect the instrument settings.

Remote Control via IEC/IEEE Bus

Setting the Device Address

The IEC/IEEE-bus address of the instrument is factory-set to 28. It can be changed manually in the Utilities - System - GPIB-Address menu or via the IEC/IEEE bus. Addresses 1 to 30 are permissible.

Manually:

- Call Utilities - System - GPIB-Address menu.
- Enter desired address.
- Terminate input using the [1x/ENTER] key.

Via IEC/IEEE bus:

```
CALL IBFIND("DEV1", generator%)                Open port to instrument
CALL IBPAD(generator%, 28)                     Transfer old address to controller
CALL IBWRT(generator%, "SYST:COMM:GPIB:ADDR 20")
                                               Set instrument to new address
CALL IBPAD(generator%, 20)                     Transfer new address to controller
```

Indications during Remote Control

The remote control state is indicated by "Remote" being displayed in the STATUS line. In the REMOTE state, the STATUS page is always displayed.
"Locked" indicates that the [LOCAL] key is disabled, ie switchover to manual control can only be made via the IEC/IEEE bus. If "Unlocked" is displayed, switchover to manual control can be made with the [LOCAL] key.

Return to Manual Operation

Return to manual operation can be made via the front panel or the IEC/IEEE bus.

Manually:

- Press [LOCAL] key.

Note:

- *Before switchover, command processing must be completed as otherwise switchover to remote control is effected immediately.*
- *The [LOCAL] key can be disabled by the universal command LLO in order to prevent unintentional switchover. In this case, switchover to manual control is only possible via the IEC/IEEE bus.*
- *The [LOCAL] key can be enabled again by deactivating the REN control line of the IEC/IEEE bus.*

Via IEC/IEEE bus:

```
...
CALL IBLOC(generator%)          Set instrument to manual control
...
```

Remote Control via RS-232-C Interface

Setting the Transmission Parameters

To enable error-free and correct data transmission, the parameters of the instrument and the controller should have the same setting. To prevent any problems during binary data transmission, the RS-232-C interface should be set to 8 data bits, “No parity” and 1 stop bit. This data format corresponds to the IEEE P1174 draft standard.
The baud rate and handshake can be manually changed in the Utilities - System - RS232 menu.

- Call Utilities – System - RS232 menu.
- Select desired baud rate and handshake.
- Terminate input using the [1x/ENTER] key.

Indications during Remote Control

The remote control state is indicated by "Remote" in the STATUS line. In the REMOTE state, the STATUS page is always displayed.

Return to Manual Operation

Return to manual operation can be made via the front panel.

- Press [*LOCAL]* key.

Note: *Before switchover, command processing must be completed as otherwise switchover to remote control is effected immediately.*

Messages

The messages transferred via the data lines of the IEC/IEEE bus can be divided into two groups:

– **interfaces messages** and
– **device messages**

No interface messages are defined for the RS-232-C interface.

Interface Messages

Interface messages are transferred on the data lines of the IEC/IEEE bus, the ATN control line being active. They are used for communication between the controller and the instrument and can only be sent by a controller which has the IEC/IEEE-bus control. Interface commands can be subdivided into

- **universal commands** and
- **addressed commands**

Universal commands act on all devices connected to the IEC/IEEE bus without previous addressing, addressed commands only act on devices previously addressed as listeners. The interface messages relevant to the instrument are listed in the section "Interface Messages" below.
Some control characters are defined for the control of the RS-232-C interface, see section "Interface Functions".

Device Messages (Commands and Device Responses)

Device messages are transferred on the data lines of the IEC/IEEE bus, the ATN control line not being active. ASCII code is used. The device messages are largely identical for the two interfaces (IEC/IEEE bus and RS-232-C).
A distinction is made according to the direction in which device messages are sent on the IEC/IEEE bus:

– **Commands** are messages the controller sends to the instrument.
They operate the device functions and request information.
Commands are subdivided according to two criteria:

1. According to the effect they have on the instrument:

Setting commands cause instrument settings such as reset of the instrument or setting the output level to 1 V.

Queries cause data to be provided for output (queries) on the IEC/IEEE bus, eg for device identification or polling of the active input.

2. According to their definition in standard IEEE 488.2:

Common Commands are exactly defined as to their function and notation in standard IEEE 488.2. They refer to functions such as the management of the standardized status registers, reset and selftest.

Device-specific commands refer to functions depending on the features of the instrument such as frequency setting. A majority of these commands has also been standardized by the SCPI committee.

– **Device responses** are messages the instruments sends to the controller in reply to a query. They may contain measurement results or information on the instrument status.

The structure and syntax of device messages are described in the following section.

Structure and Syntax of Device Messages

Introduction to SCPI

SCPI (Standard Commands for Programmable Instruments) describes a standard command set for programming instruments, irrespective of the type of instrument or manufacturer. The objective of the SCPI consortium is to standardize the device-specific commands to a large extent. For this purpose, a model was developed which defines identical functions of a device or of different devices. Command systems were generated which are assigned to these functions. Thus it is possible to address identical functions with identical commands. The command systems are of a hierarchical structure. Fig. 5-1 illustrates this tree structure using a section of command system SOURce, which operates the signal sources of the devices. The other examples concerning syntax and structure of the commands are derived from this command system.
SCPI is based on standard IEEE 488.2, ie it uses the same basic syntax elements as well as the common commands defined in this standard. Part of the syntax of the device responses is defined in greater detail than in standard IEEE 488.2 (see section "Responses to Queries").

Structure of Commands

Commands consist of a header and, in most cases, one or several parameters. The header and the parameters are separated by a "white space" (ASCII code 0 to 9, 11 to 32 decimal, eg a blank). Headers may consist of several key words. Queries are formed by appending a question mark directly to the header.

Note: *The commands used in the following examples are not in every case implemented in the instrument.*

Common commands

Common (device-independent) commands consist of a header preceded by an asterisk "*" and of one or several parameters, if any.

Examples:

`*RST`	RESET, resets the instrument
`*ESE 253`	EVENT STATUS ENABLE, sets the bits of the event status enable register
`*ESR?`	EVENT STATUS QUERY, queries the contents of the event status register

Device-specific commands

The following examples are general, they are not necessarly available with SML.

Hierarchy**:**

Device-specific commands are of a hierarchical structure (see Fig. 5-1). The different levels are represented by combined headers. Headers of the highest level (root level) have only one key word. This key word denotes a complete command system.

Example: `SOURce`
This key word denotes the `SOURce` command system.

For commands of lower levels, the complete path has to be specified, starting on the left with the highest level, the individual key words being separated by a colon ":".

Example: `SOURce:FM:EXTernal:COUPling AC`

This command is at the fourth level of the SOURce system. It selects AC coupling of the external signal source.

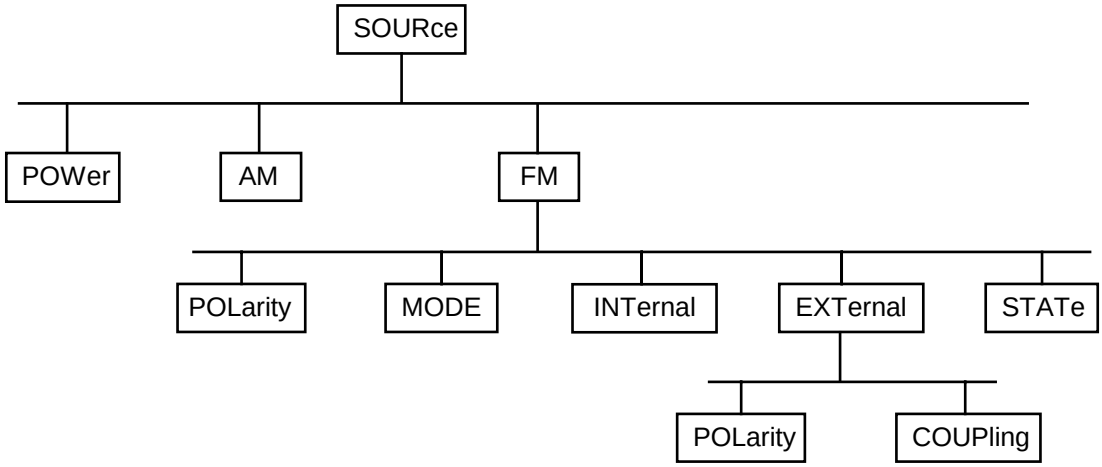


Fig. 5-1 Tree structure of SCPI command systems using the SOURce system as an example

Some key words occur at several levels within one command system. Their effect depends on the structure of the command, that is to say, at what position of the header of a command they are inserted.

Example: `:SOURce:FM:POLarity NORMal`
This command contains the key word POLarity at the third command level. It defines the polarity between the modulator and the modulation signal.

Example `:SOURce:FM:EXTernal:POLarity NORMal`
This command contains the key word POLarity at the fourth command level. It defines the polarity between the modulation voltage and the resulting direction of the modulation only for the external signal source indicated.

Optional key words:

Some command systems permit certain key words to be optionally inserted into the header or omitted. These key words are marked in the description by square brackets. The instrument must recognize the full command length for reasons of compatibility with the SCPI standard. Some commands can be considerably shortened by omitting optional key words.

Example: `[SOURce]:POWer[:LEVel][:IMMediate]:OFFSet 1`
This command immediately sets the offset of the signal to 1 dB. The following command has the same effect:

`POWer:OFFSet 1`

Note: *An optional key word must not be omitted if its effect is specified in greater detail by means of a numerical suffix.*

Long and short form:

Key words have a long form and a short form. Either the long form or the short form may be entered, other abbreviations are not permissible.

Example: `STATus:QUEStionable:ENABle 1= STAT:QUES:ENAB 1`

Note: *The short form is characterized by upper-case letters, the long form corresponds to the complete word. Upper-case and lower-case notation only serve the above purpose, the device itself does not make any difference between upper-case and lower-case letters.*

Parameters:

A parameter must be separated from the header by a "white space". If a command includes several parameters, they are separated by a comma ",". Some queries permit the parameters MINimum, MAXimum and DEFault to be entered. For a description of these parameter types see section "Parameters".

Example: `SOURce:POWer:ATTenuation? MAXimum` Response: `60`
This query requests the maximum value for the attenuation.

Numerical suffix:

If a device has several functions or features of the same kind, eg inputs, the desired function can be selected by appending a suffix to the command. Entries without suffix are interpreted like entries with the suffix 1.

Example: `SOURce2:FREQuency:MODE CW`
This command determines the operating mode for the Frequency Subsystem.

Structure of Command Lines

A command line may contain one or several commands. It is terminated by <New Line>, <New Line> with EOI or EOI together with the last data byte. QuickBASIC automatically produces EOI together with the last data byte.

Several commands in a command line are separated by a semicolon ";". If the next command belongs to a different command system, the semicolon is followed by a colon.

Example:
```
CALL IBWRT(generator%, "SOURce:POWer:CENTer MINimum;:OUTPut:ATTenuation 10")
```

This command line contains two commands. The first command belongs to the SOURce system and defines the center frequency of the output signal. The second command belongs to the OUTPut system and sets the attenuation of the output signal.

If successive commands belong to the same system and thus have one or several levels in common, the command line can be abbreviated. To this end, the second command (after the semicolon) is started with the level that lies below the common levels (see also Fig. 5-1). The colon following the semicolon must be omitted in this case.

Example:
```
CALL IBWRT(generator%, "SOURce:FM:MODE LOCKed;:SOURce:FM:INTernal:FREQuency 1kHz")
```

This command line is represented in its full length and contains two commands separated from each other by the semicolon. The two commands belong to the SOURce command system, subsystem FM, ie they have two common levels.
To abbreviate the command line, the second command is started with the level below `SOURce:FM`. The colon after the semicolon is omitted.

The abbreviated form of the command line reads as follows:

```
CALL IBWRT(generator%,  "SOURce:FM:MODE LOCKed;INTernal:FREQuency 1kHz")
```

However, a new command line always has to be started with the complete path.

Example:
```
CALL IBWRT(generator%,  "SOURce:FM:MODE LOCKed")
CALL IBWRT(generator%,  "SOURce:FM:INTernal:FREQuency 1kHz")
```

Responses to Queries

For each setting command, a query is defined unless explicitly specified otherwise. The query is formed by adding a question mark to the setting command in question. Responses to queries to the SCPI standard are partly subject to stricter rules than responses to the IEEE 488.2 standard.

1. The requested parameter is transmitted without header.
   Example: `SOURce:EXTernal:COUPling?` Response: `AC`

2. Maximum values, minimum values and all further quantities requested via a special text parameter are returned as numerical values.
   Example: `FREQuency? MAX` Response: `10E3`

3. Numerical values are output without a unit. Physical quantities are referred to the basic units or to the units set with the Unit command.
   Example: `FREQuency?` Response: `1E6` for 1 MHz

4. Truth values (Boolean parameters) are returned as 0 (for Off) and 1 (for On).
   Example: `OUTPut:STATe?` Response: `1`

5. Text (character data) is returned in a short form.
   Example: `SOURce:FM:SOURce?` Response: `INT`

Parameters

The following examples are general, they are not necessarly available with SML.
Most commands require a parameter to be specified. Parameters must be separated from the header by a "white space". Permissible parameters are numerical values, Boolean parameters, text, character strings and block data. The parameter type required for a given command and the permissible range of values are specified in the command description.

Numerical values

Numerical values can be entered in any form, ie with sign, decimal point and exponent. Values exceeding the resolution of the instrument are rounded up or down. The mantissa may comprise up to 255 characters, the exponent must be in the value range -32 000 to 32 000. The exponent is preceded by an "E" or "e". Specifying the exponent alone is not permissible. In the case of physical quantities, the unit can be entered. Permissible unit prefixes are G (giga), MA (mega, MOHM and MHz being also permissible), K (kilo), M (milli), U (micro) and N (nano). If no unit is entered, the basic unit is used.

Example: `SOURce:FREQuency 1.5 kHz = SOURce:FREQuency 1.5E3`

Special numerical values

The texts MINimum, MAXimum, DEFault, UP and DOWN are interpreted as special numerical values.
In the case of a query, the numerical value is returned.

Example: Setting command: `SOURce:VOLTage MAXimum`
Query: `SOURce:VOLTage?` Response: `15`

MIN/MAX — MINimum and MAXimum denote the minimum and the maximum value.

DEF — DEFault denotes a preset value stored in an EPROM. This value conforms to the default setting as called by the *RST command.

UP/DOWN — UP/DOWN increases or decreases the numerical value by one step. The step width can be defined via an allocated step command for each parameter which can be set via UP/DOWN (see List of Commands, chapter 6).

INF/NINF — INFinity, Negative INFinity (NINF) represent the numerical values –9.9E37 or 9.9E37, respectively. INF and NINF are only sent as device responses.

NAN — Not A Number (NAN) represents the value 9.91E37. NAN is only sent as a device response. This value is not defined. Possible causes are the division of zero by zero, the subtraction of infinite from infinite and the representation of missing values.

Boolean Parameters

Boolean parameters represent two states. The ON state (logically true) is represented by ON or a numerical value unequal to 0. The OFF state (logically untrue) is represented by OFF or the numerical value 0. In the case of a query, 0 or 1 is returned.

Example: Setting command: `SOURce:FM:STATe ON`
Query: `SOURce:FM:STATe?` Response: `1`

Text

Text parameters follow the syntactic rules for key words, ie they can be entered using a short or a long form. Like any other parameter, they must be separated from the header by a "white space". In the case of a query, the short form of the text is returned.

Example: Setting command: `:OUTPut:FILTer:TYPE EXTernal`
Query: `:OUTPut:FILTer:TYPE?` Response: `EXT`

Strings

Strings must always be entered in inverted commas (' or ").

Example: `SYSTem:LANGuage "SCPI"` or
`:SYSTem:LANGuage 'SCPI'`

Block data

Block data are a transmission format which is suitable for the transmission of large amounts of data. A command with a block data parameter has the following structure:

Example: `HEADer:HEADer #45168xxxxxxxx`

The data block is preceded by the ASCII character #. The next number indicates how many of the following digits describe the length of the data block. In the example, the four following digits indicate the length to be 5168 bytes. This is followed by the data bytes. During the transmission of the data bytes, all End or other control signs are ignored until all bytes are transmitted. Data elements comprising more than one byte are transmitted with the byte being the first which was specified by the SCPI command "`FORMat:BORDer`".

The format of the binary data within a block depends on the IEC/IEEE-bus command. The commands

```
:SOURce:CORRection:CSET:DATA:FREQuency
:SOURce:CORRection:CSET:DATA:POWer
:SYSTem:MSEQuence:DWELl
:SYSTem:MSEQuence:RCL
```

use the IEEE 754 format for double precision floating point numbers. Each number is represented by 8 bytes.

Example:

a# = 125.345678E6
b# = 127.876543E6

```
CALL IBWRT(generator%, "SOURCE:CORRECTION:CSET:DATA:FREQ
#216" + MKD$(a#) + MKD$(b#))
```

– '#' in the command string introduces the binary block,
– '2' indicates that 2 digits specifying the length will follow next,
– '16' is the length of the binary block (in bytes), here: 2 double precision floating point numbers of 8 bytes each.
– The binary data follow. Since the function IBWRT requires a text string, MKD$ is used for type conversion.

The following ASCII format has the same effect:

```
CALL IBWRT(generator%, "SOURCE:CORRECTION:CSET:DATA:FREQ
125.345678E6, 127.876543E6")
```

Overview of Syntax Elements

Following is an overview of syntax elements.

:	The colon separates the key words of a command. In a command line the separating semicolon marks the uppermost command level.
;	The semicolon separates two commands of a command line. It does not alter the path.
,	The comma separates several parameters of a command.
?	The question mark forms a query.
*****	The asterix marks a common command.
"	Quotation marks introduce a string and terminate it.
#	ASCI character # introduces block data.
	A "white space" (ASCII-Code 0 to 9, 11 to 32 decimal, e.g. blank) separates header and parameter.

Instrument Model and Command Processing

The instrument model shown in Fig. 5-2 was created with a view to the processing of IEC/IEEE-bus commands. The individual components work independently of each other and simultaneously. They communicate with each other by means of messages.

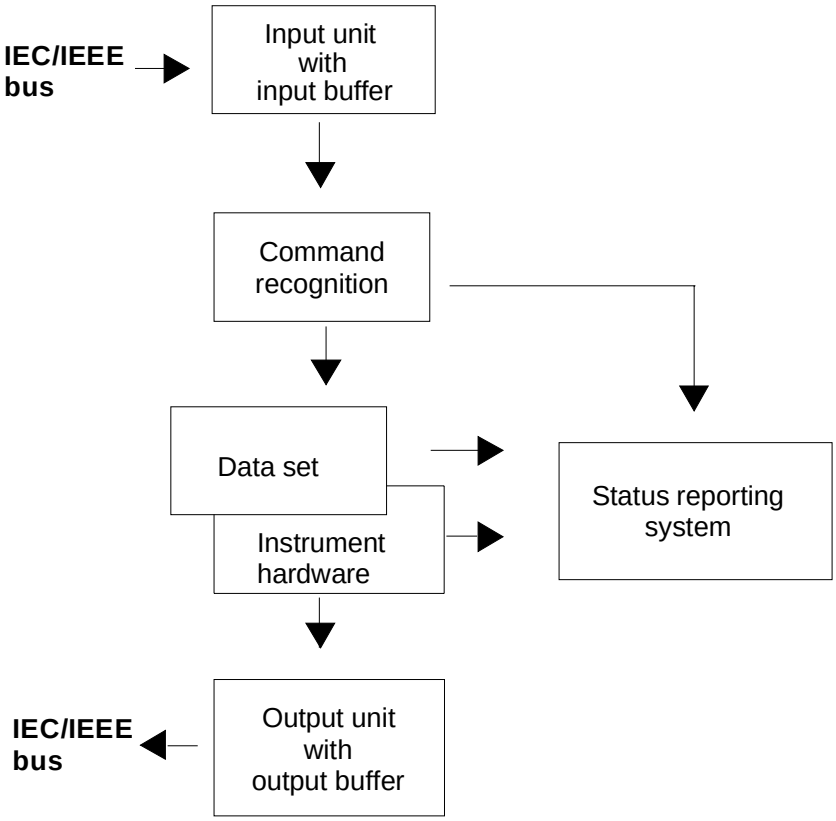


Fig. 5-2 Device model for remote control via the IEC/IEEE bus

Input Unit

The input unit receives commands character by character from the IEC/IEEE bus and stores them in the input buffer. The input buffer has a size of 256 characters. The input unit sends a message to the command recognition when the input buffer is full or when it receives a terminator, <PROGRAM MESSAGE TERMINATOR>, as defined in IEEE 488.2, or the interface message DCL.
If the input buffer is full, the IEC/IEEE-bus traffic is stopped and the data received up to then are processed. After this, the IEC/IEEE-bus traffic is continued. If, on receipt of a terminator, the input buffer is not full, the input unit can receive the next command during command recognition and execution. Receipt of a DCL command clears the input buffer and immediately initiates a message to the command recognition.

Command Recognition

The command recognition analyzes the data from the input unit in the order the data are received. Only DCL commands are serviced with priority, whereas GET commands (Group Execute Trigger), for example, are processed only after the previously received commands. Each recognized command is immediately transferred to the data set but without being executed there at once.

Syntactic errors in commands are detected here and transferred to the status reporting system. The rest of a command line following a syntax error is further analyzed and processed as far as possible.

If the command recognition recognizes a terminator or a DCL command, it requests the data set to set the commands now also in the instrument hardware. After this, it is immediately ready to continue processing commands. This means that new commands can be processed while the hardware is being set ("overlapping execution").

Data Set and Instrument Hardware

The term "instrument hardware" is used here to designate the part of the instrument which actually performs the instrument functions: signal generation, measurement, etc. The controller is not included.

The data set is a detailed reproduction of the instrument hardware in the software.

IEC/IEEE-bus setting commands cause an alteration of the data set. The data set management enters the new values (eg frequency) into the data set but passes them on to the hardware only upon request by the command recognition. As this is only effected at the end of a command line, the sequence of setting commands in the command line is not relevant.

The data are only checked for compatibility among one another and with the instrument hardware immediately before they are transferred to the instrument hardware. If it is found that an execution is not possible, an "execution error" is signalled to the status reporting system. All alterations mad to the data set are cancelled, and the instrument hardware is not reset. Due to the delayed checking and hardware setting it is permissible however that impermissible instrument states are briefly set within a command line without an error message being produced. At the end of the command line, however, a permissible instrument state must be attained.

Before the data are passed on to the hardware, the settling bit in the STATus:OPERation register is set. The hardware makes the settings and resets the bit when the new state has settled. This procedure can be used for synchronization of command processing.

IEC/IEEE-bus queries cause the data set management to send the desired data to the output unit.

Status Reporting System

The status reporting system collects information on the instrument state and makes it available to the output unit upon request. A detailed description of the structure and function is given in section "Status Reporting System".

Output Unit

The output unit collects the information requested by the controller and output by the data set management. The output unit processes the information in accordance with the SCPI rules and makes it available in the output buffer. The output buffer has a size of 256 characters. If the requested information exceeds this size, it is made available in portions without this being recognized by the controller.

If the instrument is addressed as a talker without the output buffer containing data or awaiting data from the data set management, the output unit returns the error message "Query UNTERMINATED" to the status reporting system. No data are sent on the IEC/IEEE bus. The controller waits until it has reached its time limit. This procedure is specified by SCPI.

Command Sequence and Command Synchronization

As mentioned above, overlapping execution is possible for all commands. Likewise, the setting commands of a command line are not necessarily processed in the order in which they are received.

To ensure that commands are carried out in a specific order, each command must be sent in a separate command line, ie with a separate IBWRT() call.
To prevent overlapping execution of commands, one of commands `*OPC`, `*OPC?` or `*WAI` has to be used. Each of the three commands causes a certain action to be triggered only after the hardware has been set and has settled. The controller can be programmed to wait for the respective action to occur (see Table 5-1).

Table 5-1 Synchronization by means of *OPC, *OPC? and *WAI

Command	**Action after the hardware has settled**	**Programming of controller**
*OPC	Sets the operation-complete bits in the ESR	- Setting of bit 0 in the ESE - Setting of bit 5 in the SRE – Waiting for a service request (SRQ)
*OPC?	Writes a "1" into the output buffer	Addressing of instrument as a talker
*WAI	Continues the IEC/IEEE-bus handshake. The handshake is not stopped.	Sending of next command

An example of command synchronization will be found in section 7, "Programming Examples".

Status Reporting System

The status reporting system (see Fig. 5-4) stores all information on the current operating state of the instrument, for example on any errors that have occurred. This information is stored in status registers and in an error queue. The status registers and the error queue can be queried via the IEC/IEEE bus.

The information is of a hierarchical structure. The highest level is formed by the status byte (STB) register defined in IEEE 488.2 and the associated service request enable (SRE) mask register. The STB register receives information from the standard event status register (ESR) which is also defined in IEEE 488.2 with the associated standard event status enable (ESE) mask register, and from the registers STATus:OPERation and STATus:QUEStionable which are defined by SCPI and contain detailed information on the instrument.

The status reporting system further comprises the IST flag ("Individual STatus") with the parallel poll enable (PPE) register allocated to it. The IST flag, like the SRQ, combines the entire instrument state in a single bit. The function fulfilled by the PPE register for the IST flag corresponds to that fulfilled by the SRE for the service request.

The output buffer contains the messages the instrument returns to the controller. The output buffer is not part of the status reporting system but determines the value of the MAV bit in the STB register and is therefore shown in Fig. 5-4.

Structure of an SCPI Status Register

Each SCPI register consists of five parts each of 16 bits width which have different functions (see Fig. 5-3). The individual bits are independent of each other, ie each hardware status is assigned a bit number which is valid for all five parts. For example, bit 3 of the STATus:OPERation register is assigned to the hardware status "Wait for trigger" for all five parts. Bit 15 (the most significant bit) is set to zero for all five parts. This allows the controller to process the contents of the register parts as positive integer.

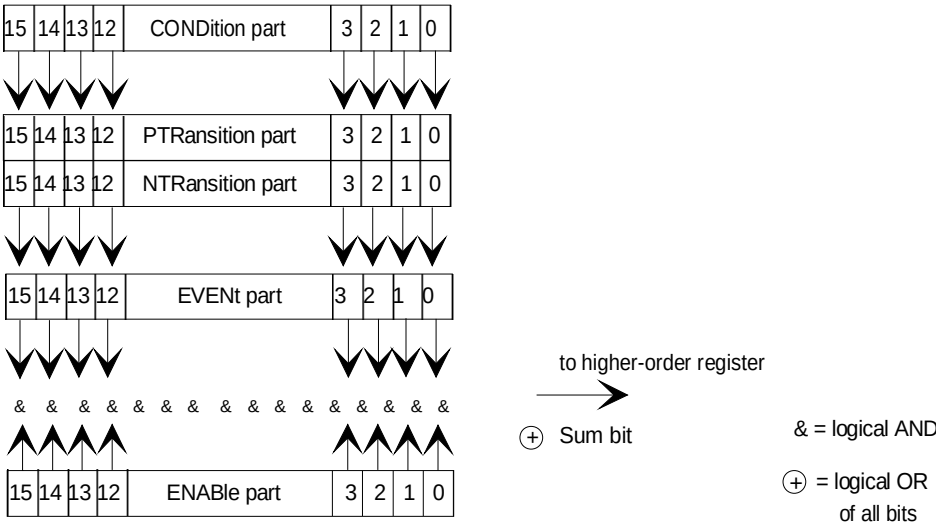


Fig. 5-3 Status register model

CONDition part The CONDition part is directly written to by the hardware or the sum bit of the next lower register. Its contents reflects the current instrument status. This register part can be read only but not written to or cleared. Reading does not affect it contents.

PTRansition part The Positive Transition part acts as an edge detector. If a bit of the CONDition part changes from 0 to 1, the status of the associated PTR bit determines whether the EVENt bit is set to 1.
PTR bit = 1: the EVENt bit is set.
PTR bit = 0: the EVENt bit is not set.
This part can be written to and read. Reading does not affect its contents.

NTRansition part The Negative Transition part likewise acts as an edge detector. If a bit of the CONDition part changes from 1 to 0, the status of the associated NTR bit determines whether the EVENt bit is set to 1.
NTR bit = 1: the EVENt bit is set.
NTR bit = 0: the EVENt bit is not set.
This part can be written to and read. Reading does not affect its contents.

With the above two edge register parts, the user can define what status transition of the CONDition part (none, 0 to 1, 1 to 0 or both) is to be stored in the EVENt part.

EVENt part The EVENt part indicates whether an event has occurred since it was read the last time; it is the "memory" of the CONDition part. It indicates only those events that were passed on by the edge filters. The EVENt part is continuously updated by the instrument. This part can be read only. Upon reading, its contents is set to zero. In linguistic usage, the EVENt part is often treated as equivalent to the complete register.

ENABle part The ENABle part determines whether the associated EVENt bit contributes to the sum bit (see below). Each bit of the EVENt part is ANDed with the associated ENABle bit (symbol '&'). The results of all logical operations of this part are passed on to the sum bit via an OR function (symbol '+').
ENABle-Bit = 0: the associated EVENt bit does not contribute to the sum bit.
ENABle-Bit = 1: if the associated EVENT bit is "1", the sum bit is set to "1" as well.
This part can be written to and read. Reading does not affect its contents.

Sum bit As mentioned above, the sum bit is obtained from the EVENt part and the ENABle part for each register. The result is entered as a bit of the CONDition part into the next higher register.
The instrument automatically generates a sum bit for each register. It is thus ensured that an event, for example a PLL that has not locked, can produce a service request throughout all hierarchical levels.

Note: *The service request enable (SRE) register defined in IEEE 488.2 can be taken as the ENABle part of the STB if the STB is structured in accordance with SCPI. Analogously, the ESE can be taken as the ENABle part of the ESR.*

Overview of Status Registers

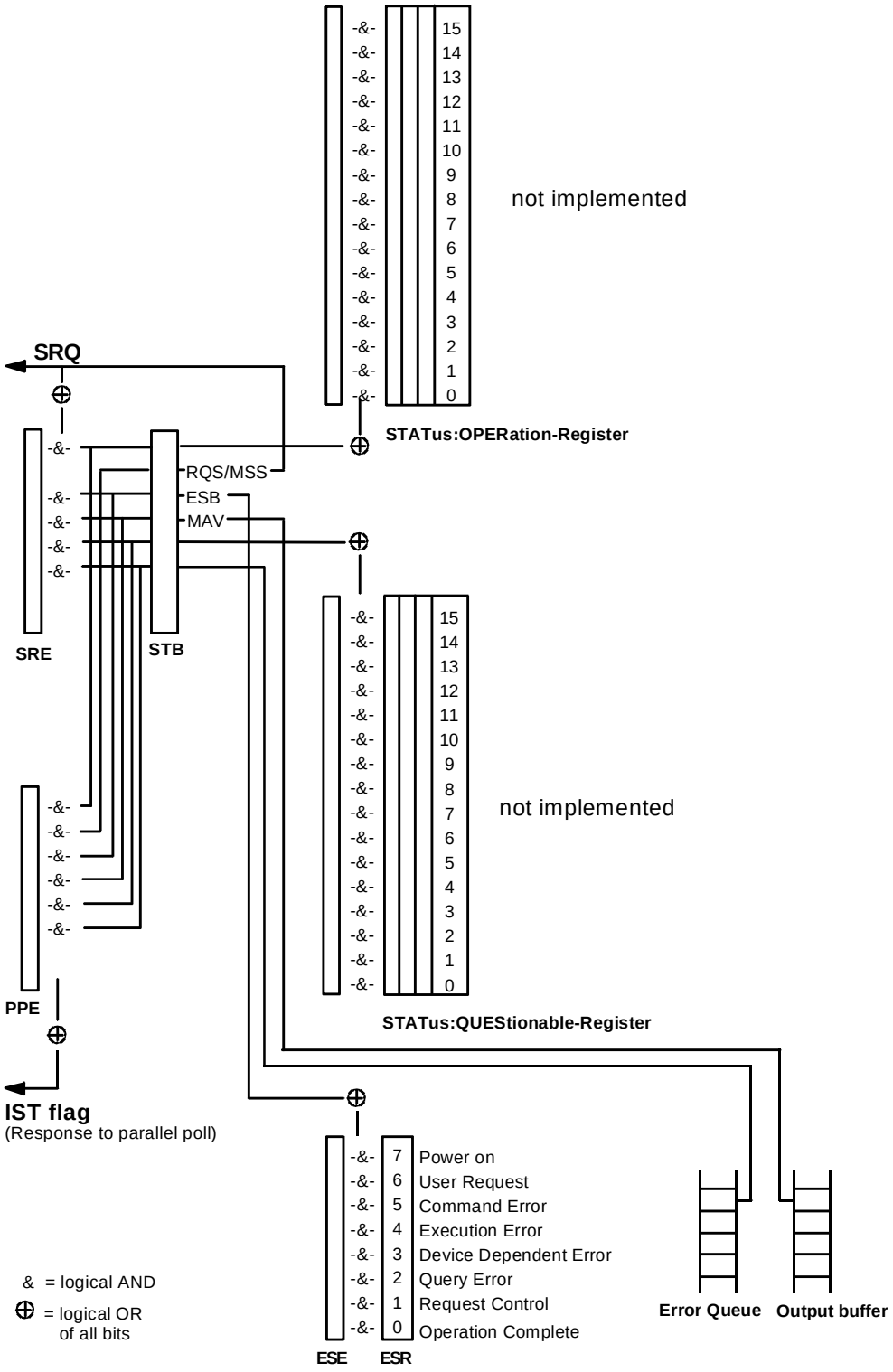


Fig. 5-4 Overview of status registers

Description of Status Registers

Status Byte (STB) and Service Request Enable Register (SRE)

The STB is already defined in IEEE 488.2. It provides a rough overview of the instrument status by collecting the pieces of information of the lower registers. It can thus be compared with the CONDition part of an SCPI register and assumes the highest level within the SCPI hierarchy. A special feature is that bit 6 acts as the sum bit of the remaining bits of the status byte.
The status byte is read using the command *STB? or a serial poll.

The STB is assigned an SRE. The SRE functionally corresponds to the ENABle part of the SCPI registers. Each bit of the STB is assigned a bit of the SRE. Bit 6 of the SRE is ignored. If a bit is set in the SRE and the associated bit in the STB changes from 0 to 1, a service request (SRQ) is generated on the IEC/IEEE bus which triggers an interrupt in the controller (if the controller is configured correspondingly) and can be further processed there.
The SRE can be set using the command *SRE and read using the command *SRE?.

Table 5-2 Meaning of the bits used in the status byte

Bit No.	Meaning
2	**Error Queue Not Empty** This bit is set if an entry is made in the error queue. If the bit is enabled by the SRE, each entry in the error queue generates a service request. Thus an error can be recognized and determined in greater detail by polling the error queue. The poll provides an informative error message. This procedure is recommended since it considerably reduces the problems involved in IEC/IEEE-bus control.
3	**QUEStionable Status sum bit** This bit is set if an EVENt bit is set in the QUEStionable status register and the associated ENABle bit is set to 1. If the bit is set, this indicates a questionable instrument status which can be determined in greater detail by polling the QUEStionable status register.
4	**MAV bit** (Message AVailable) This bit is set if a message is available in the output buffer which can be read. The bit can be used for the automatic reading of data from the instrument to the controller (see chapter 7, "Programming Examples").
5	**ESB bit** Sum bit of event status register. It is set if one of the bits of the event status register is set and enabled in the event status enable register. If the bit is set, this indicates a serious error which can be determined in greater detail by polling the event status register.
6	**MSS bit** (Master Status Summary bit) This bit is set if the instrument triggers a service request. This is the case if one of the other bits of this register is set together with its mask bit in the service request enable (SRE) register.
7	**OPERation Status Register sum bit** This bit is set if an EVENt bit is set in the OPERation status register and the associated ENABle bit is set to 1. If the bit is set, this indicates that the instrument is just carrying out an action. The type of action can be determined by polling the OPERation status register.

IST Flag and Parallel Poll Enable Register (PPE)

Analogously with the SRQ, the IST flag combines the entire status information in a single bit. It can be queried by means of a parallel poll (see section "Parallel Poll") or using the command *IST?.

The parallel poll enable (PPE) register determines which bits of the STB contribute to the IST flag. The bits of the STB are ANDed with the corresponding bits of the PPE. Unlike the SRE, bit 6 is used in this case. The IST flag results from the ORing of all results. The PPE can be set using the command *PRE and read using the command *PRE?.

Event Status Register (ESR) and Event Status Enable Register (ESE)

The ESR is already defined in IEEE 488.2. It can be compared with the EVENt part of an SCPI register. The event status register can be read using the command *ESR?.
The ESE is the associated ENABle part. It can be set using the command *ESE and read using the command *ESE?.

Table 5-3 Meaning of the bits used in the event status register

Bit No.	Meaning
0	**Operation Complete** This bit is set on receipt of the command *OPC when all previous commands have been executed.
2	**Query Error** This bit is set if either the controller wants to read data from the instrument without having sent a query, or if it does not fetch requested data and sends new instructions to the instrument instead. The cause is often a query which is errored and hence cannot be executed.
3	**Device-Dependent Error** This bit is set if a device-dependent error occurs. An error message with a number between -300 and -399 or a positive error number, which denotes the error in greater detail, is entered into the error queue (see Chapter 9, Section "Error Messages").
4	**Execution Error** This bit is set if a received command is syntactically correct but cannot be executed for other reasons. An error message with a number between -200 and –300, which denotes the error in greater detail, is entered into the error queue (see Chapter 9, Section "Error Messages").
5	**Command Error** This bit is set if a command is received which is undefined or syntactically not correct. An error message with a number between -100 and –200, which denotes the error in greater detail, is entered into the error queue (see Chapter 9, Section "Error Messages").
6	**User Request** This bit is set when the [LOCAL] key is pressed, ie when the instrument is switched over to manual control.
7	**Power On** (AC supply voltage On) This bit is set on switching on the instrument.

STATus:OPERation Register

Not impemented

STATus:QUEStionable Register

Not implemented

Use of Status Reporting System

To make effective use of the status reporting system, the information collected there must be transferred to the controller and further processed. There are several methods to this effect which are described in the following. For detailed examples see chapter 7, "Programming Examples").

Service Request, Making Use of Hierarchy Structure

Under certain conditions, the instrument can send a service request (SRQ) to the controller. The service request normally triggers an interrupt at the controller to which the control program can respond with corresponding actions. Fig. 5-4 shows that an SRQ is triggered if one or several of the bits 2, 3, 4, 5 and 7 of the status byte are set and enabled in the SRE. Each of these bits combines the information of another register, the error queue or the output buffer. By setting the ENABle parts of the status registers accordingly, it is achieved that arbitrary bits of an arbitrary status register trigger an SRQ. To make use of the possibilities of the service request, all bits of the SRE and ESE enable registers should be set to "1".

Examples (see also Fig. 5-4 and chapter 7, "Programming Examples"):

Use of command `*OPC` to generate an SRQ

- Set bit 0 in the ESE (Operation Complete).
- Set bit 5 in the SRE (ESB).

  The instrument generates an SRQ after completion of its settings.

Indication of end of sweep by means of an SRQ at the controller

- Set bit 7 (sum bit of STATus:OPERation register) in SRE.
- Set bit 3 (sweeping) in STATus:OPERation:ENABle.
- Set bit 3 in STATus:OPERation:NTRansition so that the transition of sweeping bit 3 from 1 to 0 (end of sweep) is recorded in the EVENt part.

  The instrument generates an SRQ after completion of a sweep.

The SRQ is the only way for the instrument to become active on its own. Each controller program should, therefore, set the instrument such that a service request is triggered in the event of a malfunction. The program should react appropriately to the service request. A detailed example of a service request routine is included in chapter 7, "Programming Examples".

Serial Poll

In a serial poll, just as with command `*STB`, the status byte of an instrument is queried. However, the query is implemented by means of interface messages and is therefore clearly faster. The serial-poll method has already been defined in IEEE 488.1 and used to be the only standard method for different instruments to query the status byte. The method also works with instruments which do not adhere to SCPI nor to IEEE 488.2.

The QuickBASIC command for executing a serial poll is `IBRSP( )`. Serial polling is mainly used to obtain a fast overview of the states of several instruments connected to the IEC/IEEE bus.

Parallel Poll

In a parallel poll, up to eight instruments are simultaneously requested by the controller by means of a single command to transmit 1 bit of information each on the data lines, ie to set the data line allocated to each instrument to logically "0" or "1". Analogously to the SRE register, which determines under what conditions an SRQ is generated, there is a parallel poll enable (PPE) register, which is likewise ANDed with the STB bit by bit, with bit 6 being taken into account. The results are ORed, and the result of this is sent (possibly inverted) in response to a parallel poll by the controller. The result can also be queried without a parallel poll using the command `*IST`.

The instrument first has to be set for parallel polling by means of the QuickBASIC command `IBPPC()`. This command allocates a data line to the instrument and determines whether the response is to be inverted. The parallel poll itself is executed using `IBRPP()`.

The parallel-poll method is mainly used in order to find out quickly, after an SRQ, which instrument has sent the service request if there are many instruments connected to the IEC/IEEE bus. To this effect, the SRE and the PPE must be set to the same value. A detailed example on parallel polling will be found in chapter 7, "Programming Examples".

Query by Means of Commands

Each part of every status register can be read by means of a query. The queries to be used are included with the detailed description of the registers. In response to a query, a number is always returned which represents the bit pattern of the register queried. The number is evaluated by the controller program.

Queries are normally used after an SRQ to obtain more detailed information on the cause of the SRQ.

Error Queue Query

Each error state in the instrument leads to an entry in the error queue. The entries to the error queue are detailed plain-text error messages which can be displayed in the Error menu by manual control or queried via the IEC/IEEE bus with the command `SYSTem:ERRor?`. Each call of `SYSTem:ERRor?` provides one entry from the error queue. If no more error messages are stored there, the instrument responds with 0, ie "No error".

The error queue should be queried by the controller program after each SRQ as the entries provide a more precise description of the cause of an error than the status registers. Especially during the test phase of a controller program the error queue should be queried regularly since errored commands from the controller to the instrument are also recorded in the error queue.

Reset Values of Status Reporting System

Table 5-4 lists the commands and events that cause a reset of the status reporting system. Except for *RST and SYSTem:PRESet, none of the commands has an effect on the functional settings of the instrument. It should be noted in particular that DCL also does not change instrument settings.

Table 5-4 Resetting of instrument functions

Event	Switching on of AC supply voltage		DCL, SDC (Device Clear, Selected Device Clear)	*RST or SYSTem:PRESet	STATus:PRESet	*CLS
	Power On Status Clear					
Effect	0	1				
Clears STB, ESR	—	yes	—	—	—	yes
Clears SRE, ESE	—	yes	—	—	—	—
Clears PPE	—	yes	—	—	—	—
Clears EVENt parts of the registers	—	yes	—	—	—	yes
Clears ENABle parts of all OPERation and QUESTionable registers, fills ENABle parts of all other registers with "1"	—	yes	—	—	yes	—
Fills PTRansition parts with "1", clears NTRansition parts	—	yes	—	—	yes	—
Clears error queue	yes	yes	—	—	—	yes
Clears output buffer	yes	yes	yes	1)	1)	1)
Clears command processing and input buffer	yes	yes	yes	—	—	—

1) Each command which is the first in a command line, ie which directly follows the <PROGRAM MESSAGE TERMINATOR>, clears the output buffer.

Interfaces

IEC/IEEE-Bus Interface

The instrument is equipped with an IEC/IEEE-bus interface as standard. The connector to IEEE 488 is provided at the rear of the instrument. A controller for remote control can be connected via the interface. Connection is made using a shielded cable.

Characteristics of Interface

- 8-bit parallel data transmission
- Bidirectional data transmission
- Three-wire handshake
- High data transmission rate, max. 350 kbyte/s
- Up to 15 devices can be connected
- Maximum length of connecting cables 15 m (single connection 2 m)
- Wired OR if several instruments are connected in parallel

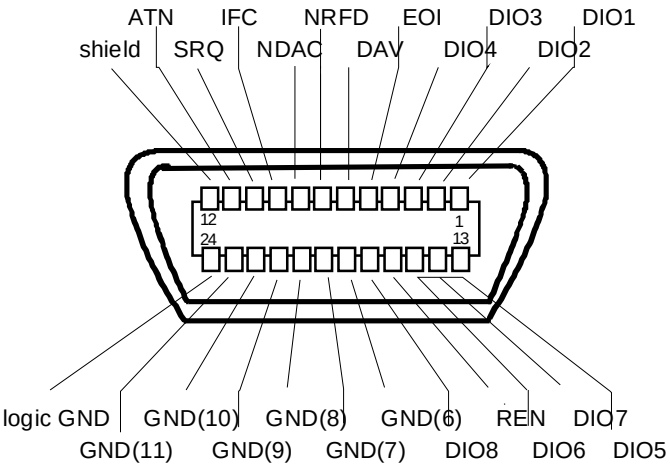


Fig. 5-5 Pin assignment of IEC/IEEE-bus interface

Bus Lines

1. Data bus with 8 lines DIO 1 to DIO 8

Transmission is bit-parallel and byte-serial in ASCII/ISO code. DIO1 is the least significant bit, DIO8 the most significant.

2. Control bus with 5 lines

IFC (Interface Clear):
Active LOW resets the interfaces of the instruments connected to the default setting.

ATN (Attention):
Active LOW signals the transmission of interface messages.
Inactive HIGH signals the transmission of device messages.

SRQ (Service Request):
Active LOW enables the instrument to send a service request to the controller.

REN (Remote Enable):
Active LOW enables switchover to remote control.

EOI (End or Identify):
This has two functions in conjunction with ATN:
ATN = HIGH Active LOW marks the end of a data transmission.
ATN = LOW Active LOW triggers a parallel poll.

3. Handshake bus with 3 lines

DAV (Data Valid):
Active LOW signals a valid data byte on the data bus.

NRFD (Not Ready For Data):
Active LOW signals that one of the devices connected is not ready to accept data.

NDAC (Not Data Accepted):
Active LOW as long as the instrument is accepting the data present on the data bus.

Interface Functions

Instruments which can be remote-controlled via the IEC/IEEE bus can be equipped with different interface functions. Table 5-5 lists the interface functions relevant for the instrument.

Table 5-5 Interface functions

Control character	Interface functions
SH1	Handshake source function (Source Handshake)
AH1	Handshake drain function (Acceptor Handshake)
L4	Listener function
T6	Talker function, ability to respond to serial poll
SR1	Service request function (Service Request)
PP1	Parallel poll function
RL1	Remote/local switchover function
DC1	Reset function (Device Clear)
DT1	Trigger function (Device Trigger)

Interface Messages

Interface messages are transmitted to the instrument on the data lines, with the ATN (Attention) line being active LOW. These messages serve for communication between the controller and the instrument.

Universal Commands

Universal commands are in the code range 10 to 1F hex. They act on all instruments connected to the bus without addressing them before.

Table 5-6 Universal commands

Command	QuickBASIC command	Effect on the instrument
DCL (Device Clear)	IBCMD (controller%, CHR$(20))	Aborts the processing of the commands just received and sets the command processing software to a defined initial state. Does not change the instrument setting.
IFC (Interface Clear)	IBSIC (controller%)	Resets the interfaces to the default state.
LLO (Local Lockout)	IBCMD (controller%, CHR$(17))	Manual switchover to LOCAL is disabled.
SPE (Serial Poll Enable)	IBCMD (controller%, CHR$(24))	Ready for serial poll.
SPD (Serial Poll Disable)	IBCMD (controller%, CHR$(25))	End of serial poll.
PPU Parallel Poll Unconfigure)	IBCMD (controller%, CHR$(21))	End of parallel polling state.

Addressed Commands

Addressed commands are in the code range 00 to 0F hex. They only act on instruments addressed as listeners.

Table 5-7 Addressed commands

Command	QuickBASIC command	Effect on the instrument
SDC (Selected Device Clear)	IBCLR (device%)	Aborts the processing of the commands just received and sets the command processing software to a defined initial state. Does not change the instrument setting.
GET (Group Execute Trigger)	IBTRG (device%)	Triggers a previously active instrument function (eg a sweep). The effect of this command is identical to that of a pulse at the external trigger signal input.
GTL (Go to Local)	IBLOC (device%)	Transition to LOCAL state (manual control).
PPC (Parallel Poll Configure)	IBPPC (device%, data%)	Configures the instrument for parallel polling. The QuickBASIC command additionally executes PPE / PPD.

RS-232-C Interface

The instrument is fitted with an RS-232-C interface as standard. The 9-contact interface is provided at the rear of the unit. A controller for remote control can be connected via the interface.

Characteristics of Interface

- Serial data transmission in asynchronous mode
- Bidirectional data transmission via two separate lines
- Selectable transmission rate from 120 to 15200 baud
- Logic 0 signal level from +3 V to +15 V
- Logic 1 signal level from –15 V to –3 V
- An external unit (controller) can be connected
- Software handshake (XON, XOFF)
- Hardware handshake

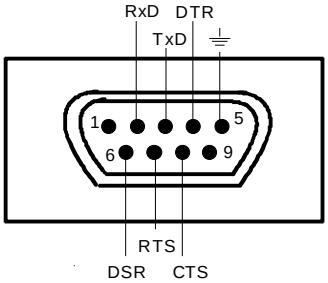


Fig. 5-6 Pin assignment of RS-232-C interface

Signal Lines

RxD (Receive Data):
Data line; transmission from external controller to instrument.

TxD (Transmit Data):
Data line; transmission from instrument to external controller.

DTR (Data terminal ready):
Output (logic zero = active). With DTR, the instrument indicates that it is ready to receive data. The DTR line controls the instrument's readiness for reception.

GND:
Interface ground, connected to instrument ground.

DSR (Data Set Ready):
(In the case of instruments with a VAR2 REV3 front module, the DSR line is used instead of the CTS line.)

RTS (Request To Send):
Output (logic 0 = active). With RTS, the instrument indicates that it is ready to receive data. The RTS line controls the instrument's readiness for reception.

CTS (Clear To Send):
Input (logic 0 = active). CTS informs the instrument that the opposite station is ready to receive data.

Transmission Parameters

To ensure error-free and correct data transmission, the transmission parameters on the instrument and the controller must have the same settings. The settings are made in the Utilities - System-RS232 menu.

Transmission rate (baud rate)
Eight different baud rates can be set on the instrument:
1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200

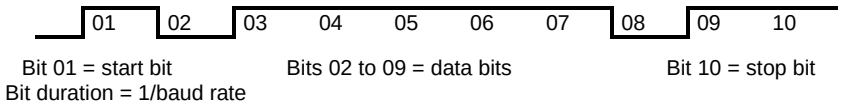
Data bits
Data transmission is in 8-bit ASCII code. The LSB (least significant bit) is transmitted as the first bit.

Start bit
The transmission of a data byte is initiated with a start bit. The falling edge of the start bit indicates the beginning of the data byte.

Parity bit
No parity bit is used.

Stop bit
The transmission of a data byte is terminated by a stop bit.

Example:
Transmission of character A (41 hex) in 8-bit ASCII code:



Interface Functions

For interface control, a number of control characters defined from 0 to 20 hex of the ASCII code can be transmitted via the interface.

Table 5-8 Control characters for RS-232-C interface

Control character	Function
<Ctrl Q> 11 hex	Enable character output (XON)
<Ctrl S> 13 hex	Stop character output (XOFF)
Break (at least 1 character logic 0)	Reset instrument
0Dhex, 0Ahex	Terminator <CR><LF> Local/remote switchover

Handshake

Software handshake
The software handshake with the XON/XOFF protocol controls data transmission.
If the receiver (instrument) wishes to inhibit the input of data, it sends XOFF to the transmitter. The transmitter then interrupts data output until it receives XON from the receiver. The same function is also provided at the transmitter end (controller).

Note: *The software handshake is not suitable for the transmission of binary data. Here the hardware handshake is to be preferred.*

Hardware handshake
With a hardware handshake, the instrument signals its readiness for reception via the lines DTR and RTS. A logic 0 means "ready", a logic 1 means "not ready".
Whether or not the controller is ready for reception is signalled to the instrument via the CTS or the DSR line (see section "Signal Lines"). The transmitter of the instrument is switched on by a logic 0 and off by a logic 1. The RTS line remains active as long as the serial interface is active. The DTR line controls the instrument's readiness for reception.

Wiring between instrument and controller
Wiring between the instrument and the controller is by means of a null modem, ie the data, control and signalling lines have to be cross-connected. The wiring plan below applies to controllers with a 9-pin or 25-pin connector.

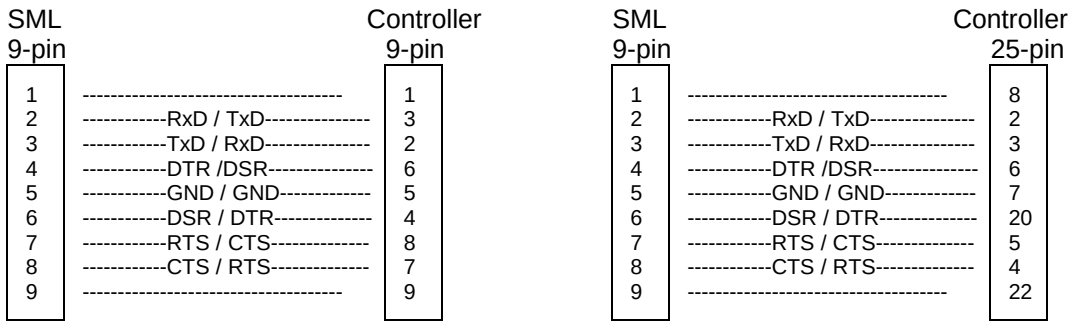


Fig. 5-7 Wiring of data, control and signalling lines for hardware handshake

6 Remote Control – Description of Commands

In the following sections, all commands implemented in the instrument are first listed in tables and then described in detail, separated according to the command system. The notation corresponds to the one of the SCPI standards to a large extent. The SCPI conformity information can be taken from the list of commands at the end of this chapter.
The description of manual operation, Chapter 4, indicates the corresponding IEC/IEEE-bus command for each manual setting.
A general introduction to remote control and a description of the status registers are to be found in Chapter 5. Detailed program examples of the main functions are to be found in Chapter 7.

Note: *In contrast to manual control, which is intended for maximum possible operating convenience, the priority of remote control is the predictability of the device status. This means that when incompatible settings are attempted, the command is ignored and the device status remains unchanged, i.e. is not adapted to other settings. Therefore, IEC/IEEE-bus control programs should always define an initial device status (e.g. with command *RST) and then implement the required settings.*

Notation

Table of Commands

Command: In the command column, the table provides an overview of the commands and their hierarchical arrangement (see indentations).

Parameter: In the parameter column the requested parameters are indicated together with their specified range.

Unit: The unit column indicates the basic unit of the physical parameters.

Remark: In the remark column an indication is made on
- whether the command does not have a query form,
- whether the command has only one query form,
- whether this command is implemented only with a certain option of the instrument.

Indentations

The different levels of the SCPI command hierarchy are represented in the table by means of indentations to the right. The lower the level is, the farther the indentation to the right is. Please observe that the complete notation of the command always includes the higher levels as well.

Example: `:SOURce:FM:MODE` is represented in the table as follows:

`:SOURce`	first level
`:FM`	second level
`:MODE`	third level

In the individual description, the complete notation of the command is given. An example for each command and - if it exists - the default value (*RST) is written out at the end of the individual description.

Upper/lower case notation

Upper/lower case letters serve to mark the long or short form of the key words of a command in the description. The instrument itself does not distinguish between upper and lower case letters.

Special characters

|

A selection of key words with an identical effect exists for several commands. These key words are indicated in the same line, they are separated by a vertical stroke. Only one of these key words has to be indicated in the header of the command. The effect of the command is independent of which of the key words is indicated.

Example: `:SOURce`
`:FREQuency`
`:CW|:FIXed`

The two following commands of identical meaning can be formed. They set the frequency of the constantly frequent signal to 9 kHz:

```
:SOURce:FREQuency:CW 9E3 = SOURce:FREQuency:FIXed 9E3
```

A vertical stroke in indicating the parameters marks alternative possibilities in the sense of "or". The effect of the command is different, depending on which parameter is entered.

Example: Selection of the parameters for the command

```
SOURce:COUPling  AC | DC
```

If parameter AC is selected, only the AC content is fed through, in the case of DC, the DC as well as the AC content.

[ ]

Key words in square brackets can be omitted when composing the header (cf. Chapter 5, Section "Optional Keywords"). The full command length must be accepted by the instrument for reasons of compatibility with the SCPI standards.
Parameters in square brackets can optionally be incorporated in the command or omitted as well.

{ }

Parameters in braces can optionally be incorporated in the command either not at all, once or several times.

Common Commands

The common commands are taken from the IEEE 488.2 (IEC 625-2) standard. Same commands have the same effect on different devices. The headers of these commands consist of an asterisk "*" followed by three letters. Many common commands refer to the status reporting system which is described in detail in Chapter 5.

Table 6-1 Common Commands

Command	Parameter	Unit	Remark
***CLS**			No query
***ESE**	0 to 255		
***ESR?**			Query only
***IDN?**			Query only
***IST?**			Query only
***OPC**			
***OPC?**			Query only
***OPT?**			Query only
***PRE**	0 to 255		
***PSC**	0 | 1		
***RCL**	1 to 50		No query
***RST**			No query
***SAV**	1 to 50		No query
***SRE**	0 to 255		
***STB?**			Query only
***TRG**			No query
***WAI**			

***CLS**

CLEAR STATUS sets the status byte (STB), the standard event register (ESR) and the EVENt-part of the QUEStionable and the OPERation register to zero. The command does not alter the mask and transition parts of the registers. It clears the output buffer.

***ESE** 0 to 255

EVENT STATUS ENABLE sets the event status enable register to the value indicated. Query *ESE? returns the contents of the event status enable register in decimal form.

***ESR?**

STANDARD EVENT STATUS QUERY returns the contents of the event status register in decimal form (0 to 255) and subsequently sets the register to zero.

***IDN?**

IDENTIFICATION QUERY queries the instrument identification.
The device response is for example: "Rohde&Schwarz,SML01,00000001,1.04"
01 = variant identification
00000001= serial number
1.04 = firmware version number

***IST?**

INDIVIDUAL STATUS QUERY returns the contents of the IST flag in decimal form (0 | 1). The IST flag is the status bit which is sent during a parallel poll.

***OPC**

OPERATION COMPLETE sets bit 0 in the event status register when all preceding commands have been executed. This bit can be used to initiate a service request.

***OPC?**

OPERATION COMPLETE QUERY returns 1, if all preceding commands have been executed. It is necessary to consider a sufficiently long time-out for the IEEE/IEC-bus.

***OPT?**

OPTION IDENTIFICATION QUERY queries the options included in the instrument and returns a list of the options installed. The options are separated from each other by means of commas. For every option, a fixed position is provided in the response.

Table 6-2 Device Response to *OPT?

Position	Option	
1	B1	Reference oscillator OXCO
2	reserved	
3	B3	Pulse modulation and pulse generator
4	reserved	
5	reserved	
6	reserved	
7	B19	Rear panel connectors

Example for a device response: B1, B3,0, 0,0,0,0,0,0, B19,0,0,0

***PRE** 0 to 255

PARALLEL POLL REGISTER ENABLE sets the parallel poll enable register to the value indicated. Query *PRE? returns the contents of the parallel poll enable register in decimal form.

***PSC** 0 | 1

POWER ON STATUS CLEAR determines whether the contents of the ENABle registers is maintained or reset in switching on.

*PSC = 0 causes the contents of the status registers to be maintained. Thus a service request can be triggered in switching on in the case of a corresponding configuration of status registers ESE and SRE.

*PSC ≠ 0 resets the registers.

Query *PSC? reads out the contents of the power-on-status-clear flag. The response can be 0 or 1.

***RCL** 1 to 50

RECALL calls the instrument state which was stored under the number supplied using command *SAV. 50 instrument states can be stored.

***RST**

RESET sets the instrument to a defined default status. The command essentially corresponds to pressing the [PRESET] key. The state of the RF-output is an exception: The RF-output is deactivated after *RST, however, it is activated after the [PRESET] key has been pressed. The default setting is indicated in the description of the commands.

***SAV** 1 to 50

SAVE stores the current instrument state under the number indicated (cf. *RCL as well).

***SRE** 0 to 255

SERVICE REQUEST ENABLE sets the service request enable register to the value indicated. Bit 6 (MSS mask bit) remains 0. This command determines under which conditions a service request is triggered. Query *SRE? reads the contents of the service request enable register in decimal form. Bit 6 is always 0.

***STB?**

READ STATUS BYTE QUERY reads out the contents of the status byte in decimal form.

***TRG**

TRIGGER triggers all actions waiting for a trigger event. Special trigger events can be started by command system "TRIGger" (see section "TRIGger System").

***WAI**

WAIT-to-CONTINUE only permits the servicing of the subsequent commands after all preceding commands have been executed and all signals have settled (cf. "*OPC" as well).

ABORt System

The ABORt system contains the commands to abort actions triggered. After an action has been aborted, it can be triggered again at once. All commands trigger an event, thus they have no *RST value.

Further commands for the trigger system of the SML can be found in the TRIGger system.

Command	Parameter	Default Unit	Remark
:ABORt			
[:SWEep]			No query

:ABORt[:SWEep]

The command restarts a sweep.

Example: :ABOR:SWE

CALibration System

The CALibration System contains the commands for external calibrations. For calibration of Ref Osc see Service Manual.

Command	Parameter	Default Unit	Remark
:CALibration			
:LEVel			
:STATe	ON | OFF		
:ATTenuator			
:STATe	ON | OFF		
:LPReset			
[:MEASure]?			
:LFGenlevel			
[:MEASure]?			
:HARMfilter			
[:MEASure]?			
:MULTfilter			
[:MEASure]?			
:IFFilter			
[:MEASure]?			
:MAINloop			
[:MEASure]?			
:FMOFfset			
[:MEASure]?			
[:ALL]			
:ROSCillator			
[:DATA]?			
:STORe			

:CALibration:LEVel:STATe

The command switches level correction ON or OFF.

Example: :CAL:LEV:STAT ON *RST value is ON

:CALibration:ATTenuator:STATe

The command switches ON or OFF the correction values of the attenuator.

Example: `:CAL:ATT:STAT ON` *RST value is ON

:CALibration:LPReset[:MEASure]?

The command calibrates Level Preset. "0" is returned for O.K. and "1" in case of an error.

Example: `:CAL:LPR?`

:CALibration:LFGenlevel[:MEASure]?

The command calibrates the level of the LF generator. "0" is returned for O.K. and "1" in case of an error.

Example: `:CAL:LFG?`

:CALibration:HARMfilter[:MEASure]?

The command calibrates the Harmonic Filters. "0" is returned for O.K. and "1" in case of an error.

Example: `:CAL:HARM?`

:CALibration:MULTfilter[:MEASure]?

The command calibrates the Multiplier Filters. "0" is returned for O.K. and "1" in case of an error.

Example: `:CAL:MULT?`

:CALibration:IFFilter[:MEASure]?

The command calibrates the IF Filters. "0" is returned for O.K. and "1" in case of an error.

Example: `:CAL:IFF?`

:CALibration:MAINloop[:MEASure]?

The command calibrates the Mainloop. "0" is returned for O.K. and "1" in case of an error.

Example: `:CAL:MAIN?`

:CALibration:FMOFfset[:MEASure]?

The command calibrates the FM offset. "0" is returned for O.K. and "1" in case of an error.

Example: `:CAL:FMOF?`

:CALibration[:ALL]?

The command executes automatically all calibrations in the indicated order. "0" is returned for O.K. and "1" in case of an error.

Example: `:CAL?`

:CALibration:ROSCillator[:DATA]?

The command displays the calibration value entered in the Utilities - Calib - RefOsc menu.

Example: `:CAL:ROSC?`

:CALibration:ROSCillator:STORe

The command stores the calibration value entered in the Utilities - Calib - RefOsc menu.

Example: `:CAL:ROSC:STOR`

DIAGnostic System

The DIAGnostic system contains the commands for diagnostic test and service of the instrument. SCPI does not define DIAGnostic commands, the commands listed here are SML-specific. All DIAGnostic commands are queries which are not influenced by *RST. Hence no default setting values are stated.

Command	Parameter	Default Unit	Remark
:DIAGnostic			
:INFO			
:CCOunt			
:POWer?			Query only
:MODules?			Query only
:OTIMe?			Query only
:SDATe?			Query only
[:MEASure]			
:POINt?			Query only

:DIAGnostic:INFO

The commands which can be used to query all information which does not require hardware measurement are under this node.

:DIAGnostic:INFO:CCOunt:POWer?

The command queries the number of switch-on processes.

Example: `:DIAG:INFO:CCO:POW?` Response: 258

:DIAGnostic:INFO:MODules?

The command queries the modules existing in the instrument with their model and state-of-modification numbers. The response supplied is a list in which the different entries are separated by commas. The length of the list is variable and depends on the equipment of the instrument. Each entry consists of three parts which are separated by means of blanks:

1. Name of module
2. Variant of module in the form VarXX (XX = 2 digits)
3. Revision of module in the form RevXX (XX = 2 digits)

Example `:DIAG:INFO:MOD?` Response: `ROSC VAR01 REV00`

:DIAGnostic:INFO:OTIMe?

The command reads out the internal operating-time counter. The response supplies the number of hours the instrument has been in operation.

Example: `:DIAG:INFO:OTIM?` Response: 19

:DIAGnostic:INFO:SDATe?

The command queries the date of software creation. The response is returned in the form year, month, day.

Example: `:DIAG:INFO:SDAT?` Response: `1999, 12, 19`

:DIAGnostic:[:MEASure]

The commands which trigger a measurement in the instrument and return the measured value are under this node.

:DIAGnostic[:MEASure]:POINt?

The command triggers a measurement at a measuring point and returns the voltage measured. The measuring point is specified by a numeric suffix (cf. service manual).

Example: `:DIAG:MEAS:POIN? 2` Response: `11.56`

DISPLAY System

This system contains the commands to configure the screen. If system security is activated using command SYSTem:SECurity ON, the display cannot be switched on and off arbitrarily (cf. below).

Command	Parameter	Default Unit	Remark
:DISPlay			
:ANNotation			
[:ALL]	ON | OFF		
:AMPLitude	ON | OFF		
:FREQuency	ON | OFF		

:DISPlay:ANNotation

The commands determining whether frequency and amplitude are indicated are under this node.

Caution: *With SYSTem:SECurity ON, the indications cannot be switched from OFF to ON. In this case *RST does not influence the ANNotation settings either. With SYSTem:SECurity OFF, the *RST value is ON for all ANNotation parameters.*

:DISPlay:ANNotation[:ALL] ON | OFF

The command switches the frequency and amplitude indication on or off.

Command :DISP:ANN:ALL ON can only be executed if SYST:SEC is set to OFF.

Example: :DISP:ANN:ALL ON With SYST:SEC OFF - *RST value is ON

:DISPlay:ANNotation:AMPLitude ON | OFF

The command switches on or off the amplitude indication.

Command :DISP:ANN:AMPL ON can only be executed if SYST:SEC is set to OFF.

Example: :DISP:ANN:AMPL ON With SYST:SEC OFF - *RST value is ON

:DISPlay:ANNotation:FREQuency ON | OFF

The command switches on or off the frequency indication.

Command :DISP:ANN:FREQ ON can only be executed if SYST:SEC is set to OFF.

Example: :DISP:ANN:FREQ ON With SYST:SEC OFF - *RST value is ON

MEMory System

This system contains the commands for the memory management of the SML.

Command	Parameter	Default Unit	Remark
:MEMory			
:NSTates?			Query only

:MEMory:NSTates?

The command returns the number of *SAV/*RCL memories available. The SML has 50 *SAV/*RCL memories in total.

Example: `:MEM:NST?` Response: `50`

OUTPut System

This system contains the commands specifying the characteristics of the RF, LF and Pulse output sockets. The following numbers are assigned to these outputs:

OUTPut1: RF output,

OUTPut2: LF output,

OUTPut3: PULSE/VIDEO output.

Command	Parameter	Default Unit	Remark
:OUTPut1|2|3			
:AMODe	AUTO | FIXed		
:POLarity			
:PULSe	NORMal | INVerted		
:SOURce	OFF | PULSegen | VIDeo		
[:STATe]	OFF | ON		
:PON	OFF | UNCHanged		
:VOLTage	0 V to 4 V	V	

:OUTPut1:AMODe AUTO | FIXed

The command switches over the operating mode of the attenuator (Attenuator MODe) at the RF output (output1).

AUTO The attenuator is switched whenever possible.

FIXed The attenuator is switched when certain fixed levels are exceeded/not reached.

Example: `:OUTP:AMOD AUTO` *RST value is AUTO

:OUTPut3:POLarity:PULSe NORMal | INVerted

The command determines the polarity of the signal at the PULSE/VIDEO output.

Example: `:OUTP3:POL:PULS INV` *RST value is NORM

:OUTPut3:SOURce OFF | PULSegen | VIDeo

The command selects between pulse generator and video output.

Example: `:OUTP3:SOUR VID` *RST value is OFF

:OUTPut1|2[:STATe] ON | OFF

The command switches on or off the RF output (output1) or the LF output (output2). The RF output can also be switched off by the response of the protective circuit. But this has no influence on this parameter.

Note: *In contrast to the PRESET key, command *RST sets the value for output1 to OFF, the RF-output is deactivated.*

Example: `:OUTP:STAT ON` *RST value is OFF

:OUTPut1[:STATe]:PON OFF | UNCHanged

This command selects the state the RF output (output1) is to assume after power-on of the unit. It only exists for the RF output. *RST does not influence the set value.

OFF Output is switched off

UNCHanged Same state as before switch-off

Example: `:OUTP:PON OFF`

:OUTPut2:VOLTage 0 V to 4 V

The command sets the voltage of the LF-output (output2). The voltage is a characteristic of the output, not of the source. I.e., the voltage is maintained even if another generator is connected to the output.

Example: `:OUTP2:VOLT 3.0V` *RST value is 1 V

SOURce System

This system contains the commands to configure the RF signal source. Keyword SOURce is optional, i.e., it can be omitted. The LF signal source is configured in the SOURce2 system.

The following subsystems are realized in the instrument:

Subsystem	Settings
[:SOURce]	
:AM	Amplitude modulation
:CORRection	Correction of the output level
:FM	Frequency modulation
:FREQuency	Frequencies including sweep
:PM	Phase modulation
:POWer	Output level, level control and level correction
:PULM	Pulse modulation
:PULSe	Pulse generator
:ROSCillator	Reference oscillator
:SWEep	Sweeps

SOURce:AM Subsystem

This subsystem contains the commands to control the amplitude modulation. An LF generator which serves as internal modulation source is fitted in the instrument. Part of the settings is effected under SOURce2.

Command	Parameters	Default Unit	Remark
[:SOURce]			
:AM			
[:DEPTh]	0 to100 PCT	PCT	
:EXTernal			
:COUPling	AC | DC		
:INTernal			
:FREQuency	0. 1 Hz to 1 MHz	Hz	
:SOURce	EXTernal | INTernal | TTONe		
:STATe	ON | OFF		

[:SOURce]:AM[:DEPTh] 0 to 100 PCT

The command sets the modulation depth in percent. *RST value is 30PCT

Example: `:SOUR:AM:DEPT 15PCT`

[:SOURce]:AM:EXTernal

The commands to set the external AM input are under this node.

[:SOURce]:AM:EXTernal:COUPling AC | DC

The command selects the type of coupling for the external AM input.

AC The d.c. voltage content is separated from the modulation signal.

DC The modulation signal is not altered. *RST value is AC

Example: `:SOUR:AM:EXT:COUP AC`

[:SOURce]:AM:INTernal

The settings for the internal AM input are effected under this node.

Here the same hardware is set for AM, FM/☞M and SOURce2. This means that, for example, the following commands are coupled to each other and have the same effect:

```
SOUR:AM:INT:FREQ
SOUR:FM:INT:FREQ
SOUR:PM:INT:FREQ
SOUR2:FREQ:CW
```

[:SOURce]:AM:INTernal:FREQuency 0.1Hz to 1 MHz

The command sets the modulation frequency.

Example: `:SOUR:AM:INT:FREQ 15kHz` *RST value is 1 kHz

[:SOURce]:AM:SOURce EXTernal **|** INTernal | TTONe

The command selects the modulation source. An external and an internal modulation source can be specified at the same time.

Example: `:SOUR:AM:SOUR EXT,INT` *RST value is INT

[:SOURce]:AM:STATe OFF | ON

The command switches amplitude modulation on or off.

Example: `:SOUR:AM:STAT ON` *RST value is OFF

SOURce:CORRection Subsystem

The CORRection subsystem permits a correction of the output level. The correction is effected by adding user-defined table values to the output level as a function of the RF frequency. In the SML, this subsystem serves to select, transmit and switch on User-Correction tables (see Section "User Correction (Ucor)" as well).

Command	Parameters	Default Unit	Remark
[:SOURce]			
:CORRection			
[:STATe]	ON | OFF		
:CSET			
:CATalog?			query only
:FREE?			query only
[:SELect]	"name of table "		
:DATA			
:FREQuency	9 kHz to F_{max} {,9 kHz to F_{max}}	Hz	F_{max} depends on model
:POWer	+20 to -20 dB {,+20 to -20 dB }	dB	
:POINts?			query only
:DELete	"name of table "		
:ALL			

[:SOURce]:CORRection[:STATe] ON | OFF

The command switches the table selected using `SOUR:CORR:CSET` on or off.

Example: `:SOUR:CORR:STAT ON` *RST value is OFF

[:SOURce]:CORRection:CSET

The commands to select and edit the Ucor tables are under this node.

[:SOURce]:CORRection:CSET:CATalog?

The command requests a list of Ucor tables. The individual lists are separated by means of commas. This command is a query and has no *RST value.

Example: `:SOUR:CORR:CAT?` Answer: `"UCOR1", "UCOR2", "UCOR3"`

[:SOURce]:CORRection:CSET:FREE?

This command queries the free space in the Ucor table.

The command is a query and thus has no *RST value.

Example: `:SOUR:CORR:FREE?`

[:SOURce]:CORRection:CSET[:SELect] "name of table"

The command selects a Ucor table. This command alone does not yet effect a correction. First the table selected must be activated (cf. `:SOUR:CORR:STAT`). If there is no table of this name, a new table is created. The name may contain up to 7 letters. This command triggers an event and hence has no *RST value.

Example: `:SOUR:CORR:CSET:SEL "UCOR1"`

[:SOURce]:CORRection:CSET:DATA

The commands to edit the Ucor tables are under this node.

[:SOURce]:CORRection:CSET:DATA:FREQuency 9 kHz to F_{max} {,9 kHz to F_{max} }, F_{max} depends on model

The command transmits the frequency data for the table selected using `:SOUR:CORR:CSET`. The frequency values must be entered in ascending order. *RST does not influence data lists.

Example: `:SOUR:CORR:CSET:DATA:FREQ 100MHz,102MHz,103MHz,...`

[:SOURce]:CORRection:CSET:DATA:POWer +20 to -20dB {,+20 to -20dB }

The command transmits the level data for the table selected using `:SOUR:CORR:CSET`. *RST does not influence data lists.

Example: `:SOUR:CORR:CSET:DATA:POWer 1dB, 0.8dB, 0.75dB,...`

[:SOURce]:CORRection:CSET:DATA:POWer:POINts?

The command returns the number of list elements.

This command is a query and hence has no *RST value.

Example: `:SOUR:CORR:CSET:DATA:POW:POIN?`

[:SOURce]:CORRection:CSET:DELete "name of table"

The command deletes the table indicated from the instrument memory. This command triggers an event and hence has no *RST value.

Example: `:SOUR:CORR:CSET:DEL "UCOR3"`

SOURce:FM Subsystem

This subsystem contains the commands to control the frequency modulation and to set the parameters of the modulation signal.

Command	Parameters	Default Unit	Remark
[:SOURce]			
:FM			
[:DEViation]	0 kHz to 20/40 MHz	Hz	
:EXTernal			
:COUPling	AC | DC		
:INTernal			
:FREQuency	0.1 Hz to 1 MHz	Hz	
:SOURce	EXTernal | INTernal | DOUBle		
:STATe	ON | OFF		
:BANDwidth	STANdard | WIDE		

[:SOURce]:FM [:DEViation] 0 kHz to 20/40 MHz

The command specifies the frequency variation caused by the FM. The maximum possible deviation depends on the selected frequency (see Data Sheet).

Example: `:SOUR:FM:DEV 5kHz` *RST value is 10 kHz

[:SOURce]:FM:EXTernal

The commands to set the external FM input are under this node. The settings under EXTernal for modulations AM and FM are independent of each other.

[:SOURce]:FM:EXTernal:COUPling AC | DC

The command selects the type of coupling for the external FM input.

AC The d.c. voltage content is separated from the modulation signal.

DC The modulation signal is not altered.

Example: `:SOUR:FM:EXT:COUP AC` *RST value is AC

[:SOURce]:FM:INTernal

The settings for the internal LF generator are effected under this node. Here the same hardware is set for AM, FM/☞M and SOURce2.

This means that, e.g., the following commands are coupled to each other and have the same effect:

```
:SOUR:AM:INT:FREQ
:SOUR:FM:INT:FREQ
:SOUR:PM:INT:FREQ
:SOUR2:FREQ:CW
```

[:SOURce]:FM:INTernal:FREQuency 0.1 Hz to 1 MHz

The command sets the modulation frequency.

Example: `:SOUR:FM:INT:FREQ 10kHz` *RST value is 1 kHz

[:SOURce]:FM:SOURce EXTernal | INTernal | TTONe

The command selects the modulation source. An external and an internal modulation source can be specified at the same time (cf. example).

Example: `:SOUR:FM:SOUR INT, EXT` *RST value is INT

[:SOURce]:FM:STATe ON | OFF

The command switches the frequency modulation on or off.

Example: `SOUR:FM:STAT OFF` *RST value is OFF

[:SOURce]:FM:BANDwidth STANdard | WIDE

The command sets the bandwidth for FM. STANdard and WIDE are available.

Example: `SOUR:FM:BAND WIDE` *RST value is STAN

SOURce:FREQuency Subsystem

This subsystem contains the commands for the frequency settings of the RF source including the sweeps.

Command	Parameters	Default Unit	Remark
[:SOURce]			
:FREQuency			
:CENTer	9 kHz to F_{max}	Hz	F_{max} depends on model
[:CW | :FIXed]	9 kHz to F_{max}	Hz	F_{max} depends on model
:RCL	INCLude | EXCLude		
:MANual	9 kHz to F_{max}	Hz	F_{max} depends on model
:MODE	CW | FIXed | SWEep		
:OFFSet	-50 to +50 GHz	Hz	
:SPAN	0 to F_{max} - 9 kHz	Hz	F_{max} depends on model
:STARt	9 kHz to F_{max}	Hz	F_{max} depends on model
:STOP	9 kHz to F_{max}	Hz	F_{max} depends on model
:STEP			
[:INCRement]	0 to 1 GHz / 0 to 2 GHz / 0 to 3 GHz	Hz	

[:SOURce]:FREQuency:CENTer 9 kHz to F_{max} (F_{max} depends on model)

The command sets the sweep range by means of the center frequency. This command is coupled to the commands `[:SOUR]:FREQ:STAR` and `[:SOUR]:FREQ:STOP`.

Here the Offset-value is taken into account.

Example: `:SOUR:FREQ:CENT 300MHz` *RST value is (STARt +STOP)/2

[:SOURce]:FREQuency[:CW | :FIXed] 9 kHz to F_{max} (F_{max} depends on model)

The command sets the frequency for CW operation. This value is coupled to the current sweep frequency. In addition to a numeric value, UP and DOWN can be indicated. The frequency is increased or reduced by the value set under `[:SOUR]:FREQ:STEP` (as to specify range, see `FREQ:CENT`).

Example: `:SOUR:FREQ 500MHz` *RST value is 100 MHz

[:SOURce]:FREQuency:RCL INCLude | EXCLude

The command determines the effect of the recall function on the frequency. *RST value has no effect to this setting.

INCLude The saved frequency is loaded when instrument settings are loaded with the [RCL] key or with a memory sequence.

EXCLude The RF frequency is not loaded when instrument settings are loaded, the current settings are maintained.

Example: `:SOUR:FREQ:RCL INCL`

[:SOURce]:FREQuency:MANual 9 kHz to F_{max} (F_{max} depends on model)

The command sets the frequency if SWE:MODE MAN and :FREQ:MODE SWE are set. Only frequency values between the settings with [:SOUR]:FREQ:STAR and ...:STOP are permitted. (As to the permitted range, cf. FREQ:CENT).

Example: :SOUR:FREQ:MAN 500MHz *RST value is 100 MHz

[:SOURce]:FREQuency:MODE CW | FIXed | SWEep

The command specifies the operating mode and hence also specifies which commands control the FREQuency subsystem. The parameters are assigned as follows:

CW | FIXed CW and FIXed are synonyms. The output frequency is specified by means of [:SOUR]:FREQ:CW | FIX.

SWEep The instrument operates in the SWEep-mode. The frequency is specified by means of commands [:SOUR]:FREQ:STAR; STOP; CENT; SPAN; MAN.

Example: :SOUR:FREQ:MODE SWE *RST value is CW

[:SOURce]:FREQuency:OFFSet -50 to +50 GHz

The command sets the frequency offset of a mixer which might be series-connected (cf. Chapter 4, Section "Frequency Offset").

Example: :SOUR:FREQ:OFFS 100MHz *RST value is 0

[:SOURce]:FREQuency:SPAN 0 to F_{max} - 9 kHz (F_{max} depends on model)

This command specifies the frequency range for the sweep. This parameter is coupled to the start and stop frequency. Negative values for SPAN are permitted, then STARt > STOP is true. The following relations hold:

STARt = CENTer - SPAN/2

STOP = CENTer + SPAN/2

Example: :SOUR:FREQ:SPAN 400MHz *RST value is (STOP - STARt)

[:SOURce]:FREQuency:STARt 9 kHz to F_{max} (F_{max} depends on model)

This command defines the starting value of the frequency for the sweep operation. Parameters STARt, STOP, SPAN and CENT are coupled to each other. STARt may be larger than STOP. (As to specified range, cf. FREQ:CENT).

Example: :SOUR:FREQ:STAR 500MHz *RST value is 100 MHz

[:SOURce]:FREQuency:STOP 9 kHz to F_{max} (F_{max} depends on model)

This command indicates the final value of the frequency for the sweep operation (see STARt as well). (As to the specified range, cf. FREQ:CENT).

Example: :SOUR:FREQ:STOP 1GHz *RST value is 500 MHz

[:SOURce]:FREQuency:STEP

The command to enter the step width for the frequency setting if frequency values UP or DOWN are used is under this node. This command is coupled to the Knob Step command in manual control. Only linear step widths can be set.

[:SOURce]:FREQuency:STEP[:INCRement] 0 to 1 GHz/0 to 2 GHz/0 to 3 GHz (SLM01/SML02/SML03)

The command sets the step width for the frequency setting.

Example: :SOUR:FREQ:STEP:INCR 1MHz *RST value is 1 MHz

SOURce:PM Subsystem

This subsystem contains the commands to control the phase modulation and to set the parameters of the modulation signal.

Command	Parameter	Default Unit	Remark
[:SOURce]			
:PM			
[:DEViation]	0 to 10	RAD	
:EXTernal			
:COUPling	AC | DC		
:INTernal			
:FREQuency	0.1 Hz to 10 MHz	Hz	
:SOURce	EXTernal | INTernal | TTONe		
:STATe	ON | OFF		
:BANDwidth	STANdard | WIDE		

[:SOURce]:PM [:DEViation] 0 to 10 RAD

The command specifies the phase variation caused by the ☞M. The maximum possible deviation depends on the selected frequency (see Data Sheet).

Example: `:SOUR:PM:DEV 2 RAD` *RST value is 1 RAD

[:SOURce]:PM:EXTernal

The commands to set the external ☞M input are under this node. The settings under EXTernal for modulations AM, FM and ☞M are independent of each other.

[:SOURce]:PM:EXTernal:COUPling AC | DC

The command selects the type of coupling for the external ☞M input.

AC The d.c. voltage content is separated from the modulation signal.

DC The modulation signal is not altered.

Example: `:SOUR:PM:EXT:COUP AC` *RST value is AC

[:SOURce]:PM:INTernal

The settings for the internal LF generator are effected under this node. Here the same hardware is set for AM, FM/☞M and SOURce2.

This means that, e.g., the following commands are coupled to each other and have the same effect:

`:SOUR:AM:INT:FREQ`

`:SOUR:FM:INT:FREQ`

`:SOUR:PM:INT:FREQ`

`:SOUR2:FREQ:CW`

[:SOURce]:PM:INTernal:FREQuency 0.1 Hz to 10 MHz

The command sets the modulation frequency.

Example: `:SOUR:PM:INT:FREQ 10kHz` *RST value is 1 kHz

[:SOURce]:PM:SOURce EXTernal | INTernal | TTONe

The command selects the modulation source. An external and an internal modulation source can be specified at the same time (cf. example).

Example: `:SOUR:PM:SOUR INT, EXT` *RST value is INT

[:SOURce]:PM:STATe ON | OFF

The command switches the phase modulation on or off.

Example: `SOUR:PM:STAT OFF` *RST value is OFF

[:SOURce]:PM:BANDwidth STANdard | WIDE

The command sets the bandwidth for ☞M. STANdard and WIDE are available.

Example: `SOUR:PM:BAND WIDE` *RST value is STAN

SOURce:POWer Subsystem

This subsystem contains the commands to set the output level, the level control and the level correction of the RF signal. Other units can be used instead of dBm:

- by indication directly after the numeric value (example `:POW 0.5V`).

Command	Parameters	Default Unit	Remark
[:SOURce]			
:POWer			
:ALC			
:SEARch?			Query only
[:STATe]	ON | OFF		
[:LEVel]			
[:IMMediate]			
[AMPLitude]	-130 dBm to +25 dBm	dBm	
:OFFSet	-100 to +100 dB	dB	
:LIMit			
[:AMPLitude]	-130 dBm to +25 dBm	dBm	
:MANual	-130 dBm to +25 dBm	dBm	
:MODE	CW | FIXed | SWEep		
:RCL	INCLude | EXCLude		
:STARt	-130 dBm to +25 dBm	dBm	
:STOP	-130 dBm to +25 dBm	dBm	
:STEP			
[:INCRement]	0.1 to 10 dB	dB	

[:SOURce]:POWer:ALC

The commands checking the automatic level control are under this node.

[:SOURce]:POWer:ALC:SEArch?

This command defines under which conditions the control loop is temporarily closed. The command is suitable only if `SOUR:POW:ALC:STAT` is set to `OFF`. This command is a query and hence has no *RST value.

Example: `:SOUR:POW:ALC:SEAR?`

[:SOURce]:POWer:ALC[:STATe] ON | OFF

The command switches the level control on or off.

ON Level control is permanently switched on.

OFF Level control is switched on for a short period of time if the level changes.

Example: `:SOUR:POW:ALC:STAT ON` *RST value is ON

[:SOURce]:POWer[:LEVel][:IMMediate]

The commands to set the output levels for the CW- and SWEEP modes are under this node.

[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude] -130 dBm to +25 dBm

The command sets the RF output level in operating mode CW. UP and DOWN can be indicated in addition to numeric values. Then the level is increased or reduced by the value indicated under [:SOUR]:POW:STEP.

In this command, the OFFSet value is considered. Thus the specified range indicated is only valid for :SOUR:POW:OFFS 0.

The keywords of this command are optional to a large extent, thus the long as well as the short form of the command is shown in the example.

Example: :SOUR:POW:LEV:IMM:AMPL -10 or

:POW –10 *RST value is -30 dBm or -20 dBm

[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:OFFSet -100 to +100 dB

The command enters the constant level offset of a series-connected attenuator/ amplifier (cf. Chapter 4, Section "Level Offset"). If a level offset is entered, the level entered using :POW does no longer conform to the RF output level. The following relation is true:

:POW = RF output level + :POW:OFFS

Entering a level offset does not change the RF output level but only the value queried by :POW. The level offset is also valid for level sweep!

Only dB is permissible as a unit here, linear units (V, W etc.) are not permitted.

Example: :SOUR:POW:LEV:IMM:AMPL:OFFS 0 or

:POW:OFFS 0 *RST value is 0 dB

[:SOURce]:POWer:LIMit[:AMPLitude] -130 dBm to +25 dBm

The command limits the maximum RF output level in operating mode CW and SWEEP. It does not influence the display LEVEL and the answer to query POW?.

Example: :SOUR:POW:LIM:AMPL 19 *RST value is +16 dBm

[:SOURce]:POWer:MANual -130 dBm to +25 dBm

The command sets the level if SOUR:POW:MODE is set to :SWE and SOUR:SWE:MODE to MAN. Only level values between STARt and STOP are permitted (as to specified range, cf. :POW:AMPL).

Example: :SOUR:POW:MAN 1dBm *RST value is -30 dBm or -20 dBm

[:SOURce]:POWer:MODE CW | FIXed | SWEep

The command specifies the operating mode and thus also by means of which commands the level setting is checked.

CW | FIXed The output level is specified by means of commands under [:SOUR]:POW:LEV.

SWEep The instrument operates in the SWEep mode. The level is specified by means of [:SOUR]:POW; STAR; STOP; CENT; SPAN and MAN.

Example: :SOUR:POW:MODE FIX *RST value is FIX

[:SOURce]:POWer:RCL INCLude | EXCLude

INCLude The stored RF level is loaded too when instrument settings are loaded.

EXCLude The stored RF level is not loaded when instrument settings are loaded ie the current level setting is maintained.

Example: :SOUR:POW:RCL INCL *RST value is EXCL

[:SOURce]:POWer:STARt -130 dBm to +25 dBm

The command sets the staring value for a level sweep. STARt may be larger than STOP, then the sweep runs from the high to the low level (As to specified range, cf. `:POW`).

Example: `:SOUR:POW:STAR -20` *RST value is -30 dBm or -20 dBm

[:SOURce]:POWer:STOP -130 dBm to +25 dBm

The command sets the final value for a level sweep. STOP may be smaller than STARt. (As to specified range, cf. `:POW`).

Example: `:SOUR:POW:STOP 3` *RST value is -10dBm

[:SOURce]:POWer:STEP[:INCRement] 0.1 to 10 dB

The command sets the step width with the level setting if UP and DOWN are used as level values. The command is coupled to Knob Step in the manual control, i.e., it also specifies the step width of the shaft encoder.

Only dB is permissible as a unit here, the linear units (V, W etc.) are not permitted.

Example: `:SOUR:POW:STEP:INCR 2` *RST value is 1dB

SOURce:PULM Subsystem

This subsystem contains the commands to control the pulse modulation (Option SML-B3) and to set the parameters of the modulation signal. The internal pulse generator is set in the `:SOURce:PULSe` subsystem.

Command	Parameters	Default Unit	Remark
[:SOURce]			
:PULM			Option SML-B3
:EXTernal			
:POLarity	NORMal | INVerse		
:SOURce	INTernal | EXTernal		
:STATe	ON | OFF		

[:SOURce]:PULM:EXTernal

The commands to control the input socket for the external pulse generator are under this node.

[:SOURce]:PULM:POLarity NORMal | INVerse

The command specifies the polarity between modulating and modulated signal.

NORMal The RF signal is suppressed during the interpulse period.

INVerse The RF signal is suppressed during the pulse.

Example: `:SOUR:PULM:POL INV` *RST value is NORM

[:SOURce]:PULM:SOURce EXTernal | INTernal

The command selects the source of the modulating signal.

INTernal Internal pulse generator.

EXTernal Signal fed externally.

Example: `:SOUR:PULM:SOUR INT` *RST value is INT

[:SOURce]:PULM:STATe ON | OFF

The command switches on or off the pulse modulation.

Example: `:SOUR:PULM:STAT ON` *RST value is OFF

SOURce:PULSe Subsystem

This subsystem contains the commands to set the pulse generator (Option SML-B3). The pulse generation is triggered on principle, with the trigger certainly being able to be set to "free run" using `TRIG:PULS:SOUR AUTO` as well.

Command	Parameters	Default Unit	Remark
[:SOURce]			
:PULSe			Option SML-B3
:DELay	20 ns to 1.3 s	s	
:DOUBle			
:DELay	60 ns to 1.3 s	s	
[:STATe]	ON | OFF		
:PERiod	100 ns to 85 s	s	
:WIDTh	20 ns to 1.3 s	s	

[:SOURce]:PULSe:DELay 20 ns to 1.3 s

The command specifies the time from the start of the period to the first edge of the pulse. Due to the construction of the instrument, this parameter is set to 0 if `[:SOUR]:PULS:DOUB:STAT` is set to `ON`. The old value is activated again as soon as the double pulse has been switched off.

Example: `:SOUR:PULS:DEL 10us` *RST value is 1 µs

[:SOURce]:PULSe:DOUBle

The commands to check the second pulse are under this node. If `[:SOUR]:PULS:DOUB:STAT` is set to `ON`, a second pulse whose width is identical to the first pulse is generated in every period.

[:SOURce]:PULSe:DOUBle:DELay 60 ns to 1.3 s

The command sets the delay time from the start of the pulse period to the first edge of the second pulse.

Example: `:SOUR:PULS:DOUB:DEL 10us` *RST value is 1 µs

[:SOURce]:PULSe:DOUBle[:STATe] ON | OFF

The command switches the second pulse on or off.

ON The second pulse is switched on.

Parameter `[:SOUR]:PULS:DEL` is set to 0 and cannot be changed. WIDTh > (`PULS:PER - PULS:DOUB:DEL`)/2 results in error message -221, "Settings conflict".

OFF The second pulse is switched off.

Example: `:SOUR:PULS:DOUB:STAT OFF` *RST value is OFF

[:SOURce]:PULSe:PERiod 100 ns to 85 s

The command sets the pulse period.

The pulse period is the reciprocal value of the pulse frequency, thus this command is coupled to command `[:SOUR]:PULM:INT:FREQ`.

Example: `:SOUR:PULS:PER 2s` *RST value is 10 µs

[:SOURce]:PULSe:WIDTh 20 ns to 1.3 s

The command sets the pulse width.

Example: `:SOUR:PULS:WIDT 0.1s` *RST value is 1 µs

SOURce:ROSCillator Subsystem

This subsystem contains the commands to set the external and internal reference oscillator.

Command	Parameters	Default Unit	Remark
[:SOURce]			
:ROSCillator			
[:INTernal]			
:ADJust			
[:STATe]	ON | OFF		
:VALue	0 to +4095		
:SOURce	INTernal | EXTernal		

[:SOURce]:ROSCillator[:INTernal]

The commands to set the internal reference oscillator are under this node.

[:SOURce]:ROSCillator[:INTernal]:ADJust

The commands for frequency adjustment (fine-tuning of the frequency) are under this node.

[:SOURce]:ROSCillator[:INTernal]:ADJust[:STATe] ON | OFF

The command switches the frequency adjustment on or off.

Example: `:SOUR:ROSC:INT:ADJ:STAT ON` *RST value is OFF

[:SOURce]:ROSCillator[:INTernal]:ADJust:VALue 0 to +4095

The command indicates the frequency correction value (tuning value). For a detailed definition, cf. Section "Reference Frequency Internal/External".

Example: `:SOUR:ROSC:INT:ADJ:VAL 0` *RST value is 0

[:SOURce]:ROSCillator[:INTernal]:RLOop NORMal | NARRow

The command sets the bandwidth of the reference loop. Normal and Narrow are available.

Example: `:SOUR:ROSC:INT:RLO NORM` *RST value is NORM

[:SOURce]:ROSCillator:SOURce INTernal | EXTernal

The command selects the reference source.

INTernal The internal oscillator is used.

EXTernal The reference signal is fed externally.

Example: `:SOUR:ROSC:SOUR EXT` *RST value is INT

SOURce:SWEep Subsystem

This subsystem contains the commands to control the RF sweep, i.e., sweeps of the RF generators. Sweeps are triggered on principle. The frequency sweep is activated by command SOUR:FREQ:MODE SWE, the level sweep by command SOUR:POW:MODE SWE.

Command	Parameters	Default Unit	Remark
[:SOURce]			
:SWEep			
[:FREQuency]			
:DWELl	10 ms to 5 s	s	
:MODE	AUTO | MANual | STEP		
:SPACing	LINear | LOGarithmic		
:STEP			
[:LINear]	0 to 1 GHz / 0 to 2 GHz / 0 to 3 GHz	Hz	SML01/SML02/SML03
:LOGarithmic	0.01 to 100 PCT	PCT	
:POWer			
:DWELl	10 ms to 5 s	s	
:MODE	AUTO | MANual | STEP		
:SPACing	LOGarithmic		
:STEP	0 to 160 dB	dB	
[:LOGarithmic]	MAXimum | MINimum		

[:SOURce]:SWEep[:FREQuency]

The commands to set the frequency sweeps are under this node. Keyword [:FREQuency] can be omitted (cf. examples). The commands are SCPI compatible then unless stated otherwise.

[:SOURce]:SWEep[:FREQuency]:DWELl 10 ms to 5 s

The command sets the dwell time per frequency step.

Example: :SOUR:SWE:DWEL 12ms *RST value is 15 ms

[:SOURce]:SWEep[:FREQuency]:MODE AUTO | MANual | STEP

The command specifies the run of the sweep.

AUTO Each trigger triggers exactly one entire sweep cycle.

MANual Each frequency step of the sweep is triggered by means of manual control or a SOUR:FREQ:MAN command, the trigger system is not active. The frequency increases or decreases (depending on the direction of the shaft encoder) by the value indicated under [:SOUR]:FREQ:STEP:INCR.

STEP Each trigger triggers only one sweep step (single-step mode). The frequency increases by the value indicated under [:SOUR]:SWE:STEP:LOG.

Example: :SOUR:SWE:MODE AUTO *RST value is AUTO

[:SOURce]:SWEep[:FREQuency]:SPACing LINear | LOGarithmic

The command selects whether the steps have linear or logarithmic spacings.

Example: `:SOUR:SWE:SPAC LIN` *RST value is LIN

[:SOURce]:SWEep[:FREQuency]:STEP

The commands to set the step width for linear and logarithmic sweeps are under this node. The settings of `:STEP:LIN` and `:STEP:LOG` are independent of each other.

[:SOURce]:SWEep[:FREQuency]:STEP[:LINear] 0 to 1 GHz/0 to 2 GHz /0 to 3 GHz (SML01/SML02/SML03)

The command sets the step width with the linear sweep. If `:STEP[:LIN]` is changed, the value of POINts valid for `:SPAC:LIN` also changes according to the formula stated under POINts. A change of SPAN does not result in a change of `:STEP[:LIN]`. Keyword `[:LIN]` can be omitted, then the command conforms to SCPI regulations (see example).

Example: `:SOUR:SWE:STEP 1MHz` *RST value is 1 MHz

[:SOURce]:SWEep[:FREQuency]:STEP:LOGarithmic 0.01 to 100 PCT

The command indicates the step width factor for logarithmic sweeps. The next frequency value of a sweep is calculated according to

new frequency = previous frequency + `STEP:LOG` x previous frequency (if STARt < STOP)

`:STEP:LOG` indicates the fraction of the previous frequency by which this is increased for the next sweep step. Usually `:STEP:LOG` is indicated in percent, with the suffix PCT having to be used explicitly. If :STEP:LOG is changed, the value of POINts valid for :SPAC:LOG also changes according to the formula stated under POINts. A change of STARt or STOP does not result in a change of :STEP:LOG.

Example: `:SOUR:SWE:STEP:LOG 10PCT` *RST value is 1 PCT

[:SOURce]:SWEep:POWer

The commands to set the power sweeps are under this node.

[:SOURce]:SWEep:POWer:DWELl 10 ms to 5 s

The command sets the dwell time per level step.

Example: `:SOUR:SWE:POW:DWEL 12ms` *RST value is 15 ms

[:SOURce]:SWEep:POWer:MODE AUTO | MANual | STEP

The command specifies the sweep mode.

AUTO Each trigger triggers exactly one entire sweep cycle.

MANual Each level step of the sweep is triggered by means of manual control or a `SOUR:POW:MAN` command, the trigger system is not active. The level increases or decreases (depending on the direction of the shaft encoder) by the value stated under `[:SOUR]:POW:STEP:INCR`.

STEP Each trigger triggers only one sweep step (single-step mode). The level increases by the value indicated under `[:SOUR]:POW:STEP:INCR`.

Example: `:SOUR:SWE:POW:MODE AUTO` *RST value is AUTO

[:SOURce]:SWEep:POWer:SPACing LOGarithmic

The command defines that the sweep steps have logarithmic spacings. It permits the query of SPACing.

Example: `:SOUR:SWE:POW:SPAC LOG` *RST value is LOG

[:SOURce]:SWEep:POWer:STEP

The commands to set the step width for the sweep are under this node.

[:SOURce]:SWEep:POWer:STEP[:LOGarithmic] 0 to 160 dB

The command indicates the step width factor for logarithmic sweeps. The next level value of a sweep is calculated according to

new level = previous level + STEP:LOG × previous level

`STEP:LOG` denotes the fraction of the previous level by which this is increased for the next sweep step. Usually `:STEP:LOG` is entered in units of dB, with suffix dB having to be specified explicitly. If `:STEP:LOG` is changed, the value of POINts also changes according to the formula indicated under POINts. A change of STARt or STOP does not result in a change of `:STEP:LOG`. Keyword `:LOG` can be omitted, then the command conforms to SCPI regulation (see example).

Example: `:SOUR:SWE:POW:STEP 10dB` *RST value is 1dB

SOURce2 System

The SOURce2 system contains the commands to configure the LF signal source. The LF signal source s designated as INT if it is used as a modulation source, if it is used as an LF generator, it is designated as SOURce2.

The commands to set the output voltage of the LF generator are in the OUTPut2 system.

Subsystems	Settings
:SOURce2	
:FREQuency	Frequency with CW and sweep operation
:SWEep	LF sweep

SOURce2:FREQuency Subsystem

This subsystem contains the commands for the frequency settings including the sweeps.

Command	Parameters	Default Unit	Remark
:**SOURce2**			
:FREQuency			
[:CW|:FIXed]	0.1 Hz to 1 MHz	Hz	
:MANual	0.1 Hz to 1 MHz	Hz	
:MODE	CW | FIXed | SWEep		
:STARt	0.1 Hz to 1 MHz	Hz	
:STOP	0.1 Hz to 1 MHz	Hz	

:SOURce2:FREQuency[:CW | :FIXed] 0.1 Hz to 1 MHz

The command sets the frequency for the CW mode.

Example: `:SOUR2:FREQ:CW 1kHz` RST value is 1 kHz

:SOURce2:FREQuency:MANual 0.1 Hz to 1 MHz

The command sets the frequency if `SOUR2:SWE:MODE MAN` and `SOUR2:FREQ:MODE SWE` are set. In this case, only frequency values between the settings `:SOUR2:FREQ:STAR` and `...:STOP` are allowed.

Example: `:SOUR2:FREQ:MAN 1kHz` *RST value is 1 kHz

:SOURce2:FREQuency:MODE CW | FIXed | SWEep

The command specifies the operating mode and hence by means of which commands the FREQuency subsystem is controlled. The following allocations are valid:

CW | FIXed CW and FIXed are synonyms. The output frequency is specified by means of `SOUR2:FREQ:CW |FIX`.

SWEep The generator operates in the SWEep mode. The frequency is specified by means of commands :`SOUR2:FREQ:STAR; STOP; MAN`.

Example: `:SOUR2:FREQ:MODE CW` *RST value is CW

:SOURce2:FREQuency:STARt 0.1 Hz to 1 MHz

This command defines the starting value of the frequency for the sweep.

Example: `:SOUR2:FREQ:STAR 1kHz` *RST value is 1 kHz

:SOURce2:FREQuency:STOP 0.1 Hz to 1 MHz

This command defines the end value of the frequency for the sweep.

Example: `:SOUR2:FREQ:STOP 200kHz` *RST value is 100 kHz

SOURce2:SWEep Subsystem

This subsystem contains the commands to control the LF sweep of SOURce2. LF-Sweeps are activated by command SOUR2:MODE SWE. Sweeps are triggered on principle.

Command	Parameters	Default Unit	Remark
:SOURce2			
:SWEep			
[:FREQuency]			
:DWELl	10 ms to 5 s	s	
:MODE	AUTO | MANual | STEP		
:SPACing	LINear | LOGarithmic		
:STEP			
[:LINear]	0 to 10MHz	Hz	
:LOGarithmic	0.01 PCT to 100 PCT	PCT	

:SOURce2:SWEep[:FREQuency]

The commands to set the frequency sweeps are under this node. Keyword [:FREQuency] can be omitted. Then the commands are SCPI-compatible unless stated otherwise (see examples).

:SOURce2:SWEep[:FREQuency]:DWELl 10 ms to 5 s

The command sets the time per frequency step (dwell).

Example: :SOUR2:SWE:DWEL 20ms *RST value is 15 ms

:SOURce2:SWEep[:FREQuency]:MODE AUTO | MANual | STEP

The command specifies the run of the sweep.

AUTO Each trigger triggers exactly one entire sweep cycle.

STEP Each trigger triggers only one sweep step (single-step mode). The frequency increases by the value defined under :SOUR2:SWE:STEP.

Example: :SOUR2:SWE:MODE AUTO *RST value is AUTO

:SOURce2:SWEep[:FREQuency]:SPACing LINear | LOGarithmic

The command selects whether the steps have linear or logarithmic spacings.

Example: :SOUR2:SWE:SPAC LOG *RST value is LIN

:SOURce2:SWEep[:FREQuency]:STEP

The commands to set the step width with linear and logarithmic sweeps are under this node. The settings of STEP:LIN and STEP:LOG are independent of each other.

:SOURce2:SWEep[:FREQuency]:STEP[:LINear] 0 to 1 MHz

The command sets the step width with the linear sweep. If STEP:LIN is changed, the value of POINts valid for SPAC:LIN also changes according to the formula defined under POINts. A change of SPAN does not cause a change of STEP:LIN. Keyword [:LIN] can be omitted, then the command conforms to SCPI regulation (see example).

Example: :SOUR2:SWE:STEP 10kHz *RST value is 1 kHz

:SOURce2:SWEep[:FREQuency]:STEP:LOGarithmic 0.01 to 100PCT

This command defines the step width factor for logarithmic sweeps. The next frequency value of a sweep is calculated as follows (if STARt < STOP) :

New frequency = previous frequency + STEP:LOG x previous frequency

STEP:LOG , therefore, indicates the fraction of the previous frequency by which that frequency is increased for the next sweep step. STEP:LOG is usually indicated in percent, with the suffix PCT having to be used explicitly. If STEP:LOG is changed, the value of POINts valid for SPACing:LOGarithmic also changes according to the formula stated under POINts. A change of STARt or STOP does not result in a change of STEP:LOGarithmic.

Example: :SOUR2:SWE:STEP:LOG 5PCT *RST value is 1 PCT

STATus System

This system contains the commands for the status reporting system (c.f. Section "Status Reporting System"). STATus:OPERation register and STATus:QUEStionable register are not implemented. *RST has no influence on the status registers.

Command	Parameters	Default Unit	Remark
:STATus			
:PRESet			No query
:QUEue			
[:NEXT]?			Query only

:STATus:PRESet

The command resets the edge detectors and ENABle parts of all registers to a defined value. All PTRansition parts are set to FFFFh, i.e., all transitions from 0 to 1 are detected. All NTRansition parts are set to 0, i.e., a transition from 1 to 0 in a CONDition bit is not detected.

Example: `:STAT:PRES`

:STATus:QUEue [:NEXT]?

The command queries the entry that has been in the error queue for the longest time and thus deletes it. Positive error numbers denote errors specific of the instrument, negative error numbers error messages specified by SCPI (see Chapter 5). If the error queue is empty, 0, "No error", is returned. The command is identical to SYST:ERR?

Example: `STAT:QUE:NEXT?` Answer: `221, "Settings conflict"`

SYSTem System

In this system, a number of commands for general functions which are not immediately related to signal generation, are combined.

Command	Parameters	Default Unit	Remark
:SYSTem			
:COMMunicate			
:GPIB			
[:SELF]			
:ADDRess	1 to 30		
:SERial			
:BAUD	1200 | 2400 | 4800 | 9600 | 19200 | 38400 | 57600 | 115200		
:BITS	7 | 8		
:SBITs	1 | 2		
:CONTrol			
:RTS	ON | IBFull | RFR		
:PACE	XON | NONE		
:PARity	ODD | EVEN | NONE		
:DISPlay			
:UPDate			
[:STATe]	ON | OFF		
:ERRor?			Query only
:PRESet			No query
:PROTect[1|2|3|4]			
[:STATe]	ON | OFF, password		
:SECurity			
[:STATe]	ON | OFF		
:SERRor?			Query only
:VERSion?			Query only

:SYSTem:COMMunicate

The commands to set the remote control interfaces are under this node.

:SYSTem:COMMunicate:GPIB

The commands to control the IEC bus are under this node (GPIB = **G**eneral **P**urpose **I**nterface **B**us).

:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess 1 to 30

The command sets the IEC bus instrument address. *RST value is 28

Example: `:SYST:COMM:GPIB:ADDR 1`

:SYSTem:COMMunicate:SERial

The command to set the serial interface are under this node. The data format is fixedly set to 8 data bits, no parity and 1 stop bit. These values cannot be changed. The device represents a DTE (Data Terminal Equipment) in relation to the serial interface. Therefore the the controller must be connected via a 0-modem.

:SYSTem:COMMunicate:SERial:BAUD 1200| 2400| 4800| 9600| 19200| 38400| 57600| 115200

The commands sets the baud rate for both the transmit and the receive direction. *RST has no influence on this parameter.

Example: `:SYST:COMM:SER:BAUD 1200` *RST value is 9600

:SYSTem:COMMunicate:SERial:BITS 7 | 8

The command sets the length of a data word.

Example: `:SYST:COMM:SER:BITS` *RSTvalue is 7

:SYSTem:COMMunicate:SERial:SBITs 1 | 2

The command defines whether 1 or 2 stop bits are used.

Example: `:SYST:COMM:SER:SBIT` *RST value is 1

:SYSTem:COMMunicate:SERial:CONTrol:RTS ON | IBFull | RFR

The commands sets the hardware handshake. *RST has no influence on this parameter.

ON Interface line RTS is always active.

IBFull | RFR Input Buffer Full | Ready For Receiving.
Interface line RTS remains active as long as the instrument is ready to receive data.

Example: `:SYST:COMM:SER:CONT:RTS ON` *RST value is RFR

:SYSTem:COMMunicate:SERial:PACE XON | NONE

The command sets the software handshake. *RST has no influnence on this parameter.

XON Software handshake using the ASCII codes 11h (XON) and 13h (XOFF).

Note: *This mode is not recommended for binary data and for baud rates above 9600 bauds.*

NONE No software handshake.

Example: `:SYST:COMM:SER:PACE NONE` *RST value is NONE

:SYSTem:COMMunicate:SERial:PARity ODD | EVEN | NONE

The command defines the parity test.

Example: `:SYST:COMM:SER:PAR ODD` *RST value is EVEN

:SYSTem:DISPlay:UPDate[:STATe] ON | OFF

ON The header line of the display indicates frequency and level values.

OFF The header line of the display remains empty.

This function is only available via IEC/IEEE-bus.

Example: `:SYST:DISP:UPD OFF` *RST value is ON

:SYSTem:ERRor?

The command queries the entry that has been in the error queue for the longest time. Positive error numbers denote errors specific of the instrument, negative error numbers denote error messages specified by SCPI (see Chapter 9). If the error queue is empty, 0, "No error", is returned. The command is identical to `STAT:QUE:NEXT?`

Example: `:SYST:ERR?` Answer: `-221, "Settings conflict"`

:SYSTem:PRESet

The command triggers an instrument reset. It has the same effect as the PRESET key of the manual control or as command *RST. This command triggers an event and hence has no *RST value.

Example: `:SYST:PRES`

:SYSTem:PROTect[1|2|3|4]

The command to disable certain instrument functions is under this node. A list of the functions concerned can be found in the manual control (Chapter 4, Section "Password Input With Protected Functions"). There are four protection levels which are distinguished by means of a suffix after PROT. *RST has no effects on the disabling/enabling of the instrument functions.

:SYSTem:PROTect[1|2|3|4][:STATe] ON | OFF, Password

The command switches a protection level on or off. The passwords are 6-digit numbers. They are fixedly stored in the firmware. The password for the first level is 123456.

ON disables the functions belonging to this protection level. A password doesn't have to be entered.

OFF deactivates the disabling again if the correct password is entered. Otherwise an error -224, "Illegal parameter value" is generated and STATe remains ON.

Example: `:SYST:PROT1:STAT OFF, 123456`

:SYSTem:SECurity[:STATe] ON | OFF

The command switches the security state on or off.

ON The following commands cannot be executed:
`:DISP:ANN:ALL ON`
`:DISP:ANN:FREQ ON`
`:DISP:ANN:AMPL ON`

OFF In the transition from ON to OFF all data existing in the instrument except for the calibrating data are deleted, especially all status registers, all instrument states and all lists.

The command is not influenced by *RST and *RCL.

Example: `:SYST:SEC:STAT ON`

:SYSTem:SERRor?

This command returns a list of all errors existing at the point of time of the query. The error messages are separated by commas. This list corresponds to the indication on the ERROR page with manual control (cf. Chapter 9, Section "Error Messages").

Example: `:SYST:SERR?`
Answer: `-221, "Settings conflict", 153, "Input voltage out of range"`

:SYSTem:VERSion?

The command returns the SCPI version number the instrument acts in accordance with. This command is a query and thus has no *RST value.

Example: `:SYST:VERS?` Answer: `1994.0`

TEST System

This system contains the commands to execute the selftest routines (RAM?, ROM? and BATT?) as well as to directly manipulate the hardware modules (:TEST:DIR). The selftests return a "0" if the test has been executed successfully, otherwise a value unequal to "0". All commands of this system do not have an *RST value.

Caution: *The commands under node :TEST:DIR directly act on the respective hardware module circumventing any security mechanisms. They are provided for service purposes and should not be used by the user. Improper use of the commands may damage the module.*

Command	Parameters	Default Unit	Remark
:TEST			
:DIRect	Address, subaddress, hex data string		
:ASSy	Module, subaddress, hex data string		
:RAM?			Query only
:ROM?			Query only
:BATTery?			Query only

:TEST:DIRect Address, subaddress, hex data string

This node contains the commands directly acting on the respective hardware module circumventing any security mechanisms. The commands under this node have no short form.

:TEST:ASSy Module, subaddress, hex data string

This command addresses the ASSy module. A subaddress (0 or 1) must be entered as a parameter. The data are entered as a <string> (ie an ASCII character string enclosed in inverted commas) representing hexadecimal numbers. The string, therefore, may contain the characters 0 to 9 A to F.

:TEST:RAM?

The command triggers a test of the RAM.

:TEST:ROM?

The command triggers a test of the main memory (EEPROM).

:TEST:BATTery?

The command triggers a test of the battery voltage.

TRIGger System

The TRIGger system contains the commands to select the trigger source and to configure the external trigger socket. The trigger sources for the individual signal sources (RF, LFGen) are distinguished by a numerical suffix appended to TRIG. The suffix conforms to the numbering of the SOURce system:

TRIGger1 = RF generator

TRIGger2 = LFGen

The trigger system of the SML consists of a simplified implementation of the SCPI trigger system. Compared to SCPI, the TRIGger system shows the following differences:

- No INIT command, the instrument behaves as if `:INIT:CONT ON` was set.
- There are several subsystems denoting the different parts of the instrument under TRIGger (SWEep, PULSe).

Further commands as to the trigger system of the SML can be found in the ABORt system.

Command	Parameters	Default Unit	Remark
:TRIGger1|2			
[:SWEep]			
[:IMMediate]			No query
:SOURce	SINGle | EXTernal | AUTO		
:PULSe			
:EGATed			
:POLarity	NORMal | INVerted		
:SOURce	AUTO | EXTernal | EGATed		
:SLOPe	POSitive | NEGative		

:TRIGger1|2[:SWEep]

All commands to trigger a sweep are under this node. The settings here act on level and frequency sweeps for RF generator (TRIG1) or LF generator (TRIG2).

:TRIGger1|2[:SWEep][:IMMediate]

The command immediately starts a sweep. Which sweep is executed depends on the respective Mode setting, e.g. `:SOUR:FREQ:MODE SWE`. The command corresponds to manual-control command Execute Single Sweep. This command triggers an event and thus has no *RST value.

Example: `:TRIG:SWE:IMM`

:TRIGger1|2[:SWEep]:SOURce AUTO | SINGle | EXTernal

The command specifies the trigger source. The naming of the parameters directly corresponds to the different settings with manual control. SCPI uses other designations for the parameters the instrument accepts as well. These designations are to be preferred if compatibility is important. The following table provides an overview.

SML designation	SCPI designation	Command with manual control
AUTO	IMMediate	Mode Auto
SINGle	BUS	Mode Single or Step
EXTernal	EXTernal	Mode Ext Trig Single or Ext Trig Step

AUTO The trigger is free-running, i.e., the trigger requirement is permanently met. As soon as a sweep has been terminated, the next one is started.

SINGle Triggering is effected by means of IEC-bus commands `:TRIG:SWE:IMM` or `*TRG`. If `:SOUR:SWE:MODE` is set to `STEP`, a step, in the case of the `AUTO` setting a complete sweep, is executed.

EXTernal Triggering is effected from outside via the TRIGGER socket or by the GET command via IEC/IEEE-bus. The action triggered depends on the setting of the sweep mode as in the case of SINGle.

Example: `:TRIG:SWE:SOUR AUTO` *RST value is SING

:TRIGger:PULSe

This node contains all commands to trigger the pulse generator (Option SML-B3). The commands are only valid for TRIGger1.

:TRIGger:PULSe:EGATed:POLarity NORMal | INVerted

The command defines the active level of the gate signal.

NORMal Active level = HIGH
INVerted Active level = LOW

Example: `:TRIG:PULS:EGAT:POL INV` *RST value is NORM

:TRIGger:PULSe:SOURce AUTO | EXTernal | EGATed

The command specifies the trigger source.

AUTO Trigger is free-running (see above).

EXTernal Triggering is effected from outside via the PULSE socket.

EGATed Triggering is effected when the gate signal is active.

Example: `:TRIG:PULS:SOUR AUTO` *RST value is AUTO

:TRIGger:PULSe:SLOPe POSitive | NEGative

The command defines whether the action triggered is triggered at the positive or the negative edge of the trigger signal.

Example: `:TRIG:PULS:SLOP NEG` *RST value is POS

List of Commands

Command	Parameter	SCPI info	Page
:ABORt[:SWEep]		not-SCPI	6.6
:CALibration:LEVel:STATe	ON | OFF	not SCPI	6.6
:CALibration:ATTenuator	ON | OFF	not SCPI	6.7
:CALibration:LPReset[:MEASure]?		not SCPI	6.7
:CALibration:LFGenlevel[:MEASure]?		not SCPI	6.7
:CALibration:HARMfilter[:MEASure]?		not SCPI	6.7
:CALibration:MULTfilter[:MEASure]?		not SCPI	6.7
:CALibration:IFFilter[:MEASure]?		not SCPI	6.7
:CALibration:MAINloop[:MEASure]?		not SCPI	6.7
:CALibration:FMOFfset[:MEASure]?		not-SCPI	6.7
:CALibration[:ALL?]		not SCPI	6.7
:DIAGnostic:INFO:CCOunt:POWer?		not-SCPI	6.8
:DIAGnostic:INFO:MODules?		not-SCPI	6.8
:DIAGnostic:INFO:OTIMe?		not-SCPI	6.8
:DIAGnostic:INFO:SDATe?		not-SCPI	6.9
:DIAGnostic[:MEASure]:POINt?		not-SCPI	6.9
:DISPlay:ANNotation[:ALL]	ON | OFF		6.10
:DISPlay:ANNotation:AMPLitude	ON | OFF		6.10
:DISPlay:ANNotation:FREQuency	ON | OFF		6.10
:MEMory:NSTates?			6.11
:OUTPut1:AMODe	AUTO | FIXed	not-SCPI	6.11
:OUTPut3:POLarity:PULSe	NORMal | INVerse		6.12
:OUTPut3:POLarity:VIDeo	NORMal | INVerse		6.12
:OUTPut3:SOURce	OFF | PULSegen | VIDeo		6.12
:OUTPut1|2[:STATe]	ON | OFF		6.12
:OUTPut1[:STATe]:PON	OFF | UNCHanged	not-SCPI	6.12
:OUTPut2:VOLTage	0 V to 4 V	not-SCPI	6.12
[:SOURce]:AM[:DEPTh]	0 to 100 PCT		6.13
[:SOURce]:AM:EXTernal:COUPling	AC | DC		6.14
[:SOURce]:AM:INTernal:FREQuency	0.1 Hz to 10 MHz		6.14
[:SOURce]:AM:SOURce	EXTernal | INTernal | TTONe		6.14
[:SOURce]:AM:STATe	OFF | ON		6.14
[:SOURce]:CORRection[:STATe]	ON | OFF		6.15
[:SOURce]:CORRection:CSET:CATalog?		not-SCPI	6.15
[:SOURce]:CORRection:CSET:FREE?		not-SCPI	6.15
[:SOURce]:CORRection:CSET[:SELect]	'name of table'		6.16
[:SOURce]:CORRection:CSET:DATA:FREQuency	9 kHz to 1.1 GHz {,9 kHz to 1.1 GHz }	not-SCPI	6.16
[:SOURce]:CORRection:CSET:DATA:POWer	+20 to -20dB {,+20 to -20dB }	not-SCPI	6.16
[:SOURce]:CORRection:CSET:DATA:POWer:POINts?		not-SCPI	6.16
[:SOURce]:CORRection:CSET:DELete	'name of table'	not-SCPI	6.16
[:SOURce]:FM[:DEViation]	0 kHz to 20/40 MHz	not-SCPI	6.17
[:SOURce]:FM:EXTernal:COUPling	AC | DC		6.17
[:SOURce]:FM:INTernal:FREQuency	0.1 Hz to 10 MHz		6.18

Command	Parameter	SCPI info	Page
[:SOURce]:FM:SOURce	EXTernal | INTernal | TTONe		6.18
[:SOURce]:FM:STATe	ON | OFF		6.18
[:SOURce]:FM:BANDwidth	STANdard | WIDE		6.18
[:SOURce]:FREQuency:CENTer	9 kHz to 1.1 GHz		6.19
[:SOURce]:FREQuency[:CW | :FIXed]	9 kHz to 1.1 GHz		6.19
[:SOURce]:FREQuency:RCL	INCLude | EXCLude		6.19
[:SOURce]:FREQuency:MANual	9 kHz to 1.1 GHz		6.20
[:SOURce]:FREQuency:MODE	CW | FIXed | SWEep		6.20
[:SOURce]:FREQuency:OFFSet	-50 to +50 GHz		6.20
[:SOURce]:FREQuency:SPAN	1.1 GHz – 9 kHz		6.20
[:SOURce]:FREQuency:STARt	9 kHz to 1.1 GHz		6.20
[:SOURce]:FREQuency:STOP	9 kHz to 1.1 GHz		6.20
[:SOURce]:FREQuency:STEP[:INCRement]	0 to 1 GHz		6.20
[:SOURce]:PM[:DEViation]	0 to 10 RAD	not-SCPI	6.21
[:SOURce]:PM:EXTernal:COUPling	AC | DC		6.21
[:SOURce]:PM:INTernal:FREQuency	0.1 Hz to 10 MHz		6.22
[:SOURce]:PM:SOURce	EXTernal | INTernal | TTONe		6.22
[:SOURce]:PM:STATe	ON | OFF		6.22
[:SOURce]:PM:BANDwidth	STANdard | WIDE		6.22
[:SOURce]:POWer:ALC:SEArch?			6.23
[:SOURce]:POWer:ALC[:STATe]	ON | OFF		6.23
[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]	–130 dBm to +25 dBm		6.24
[:SOURce]:POWer[:LEVel][:IMMediate][:AMPLitude]:OFFSet	-100 to +100 dB		6.24
[:SOURce]:POWer:LIMit[:AMPLitude]	-130 dBm to +25 dBm		6.24
[:SOURce]:POWer:MANual	–130 dBm to +25 dBm		6.24
[:SOURce]:POWer:MODE	CW | FIXed | SWEep		6.24
[:SOURce]:POWer:STARt	-130 dBm to +25 dBm		6.25
[:SOURce]:POWer:STOP	-130 dBm to +25 dBm		6.25
[:SOURce]:POWer:STEP[:INCRement]	0.1 to 10 dB		6.25
[:SOURce]:PULM:POLarity	NORMal | INVerse		6.26
[:SOURce]:PULM:SOURce	EXTernal | INTernal		6.26
[:SOURce]:PULM:STATe	ON | OFF		6.26
[:SOURce]:PULSe:DELay	20 ns to 1.3 s		6.27
[:SOURce]:PULSe:DOUBle:DELay	60 ns to 1.3 s		6.27
[:SOURce]:PULSe:DOUBle[:STATe]	ON | OFF		6.27
[:SOURce]:PULSe:PERiod	100 ns to 85 s		6.27
[:SOURce]:PULSe:WIDTh	20 ns to 1.3 s		6.27
[:SOURce]:ROSCillator[:INTernal]:ADJust[:STATe]	ON | OFF	not-SCPI	6.28
[:SOURce]:ROSCillator[:INTernal]:ADJust:VALue	0 to +4095	not-SCPI	6.28
[:SOURce]:ROSCillator[:INTernal]:RLOop	NORMal | NARRow	not-SCPI	6.28
[:SOURce]:ROSCillator:SOURce	INTernal | EXTernal		6.28
[:SOURce]:SWEep[:FREQuency]:DWELl	10 ms to 5 s	not-SCPI	6.29
[:SOURce]:SWEep[:FREQuency]:MODE	AUTO | MANual | STEP	not-SCPI	6.29
[:SOURce]:SWEep[:FREQuency]:SPACing	LINear | LOGarithmic	not-SCPI	6.30
[:SOURce]:SWEep[:FREQuency]:STEP[:LINear]	0 to 1 GHz	not-SCPI	6.30

Command	Parameter	SCPI info	Page
[:SOURce]:SWEep[:FREQuency]:STEP:LOGarithmic	0.01 to 10PCT	not-SCPI	6.30
[:SOURce]:SWEep:POWer:DWELl	10 ms to 5 s	not-SCPI	6.30
[:SOURce]:SWEep:POWer:MODE	AUTO | MANual | STEP	not-SCPI	6.30
[:SOURce]:SWEep:POWer:SPACing	LOGarithmic	not-SCPI	6.31
[:SOURce]:SWEep:POWer:STEP[:LOGarithmic]	0 to 160 dB	not-SCPI	6.31
:SOURce2:FREQuency[:CW | :FIXed]	0.1 Hz to 1 MHz		6.32
:SOURce2:FREQuency:MANual	0.1 Hz to 1 MHz		6.32
:SOURce2:FREQuency:MODE	CW | FIXed | SWEep		6.33
:SOURce2:FREQuency:STARt	0.1 Hz to 1 MHz		6.33
:SOURce2:FREQuency:STOP	0.1 Hz to 1 MHz		6.33
:SOURce2:SWEep[:FREQuency]:DWELl	10 ms to 5 s	not-SCPI	6.34
:SOURce2:SWEep[:FREQuency]:MODE	AUTO | MANual | STEP	not-SCPI	6.34
:SOURce2:SWEep[:FREQuency]:SPACing	LINear | LOGarithmic	not-SCPI	6.34
:SOURce2:SWEep[:FREQuency]:STEP[:LINear]	0 to 1 MHz	not-SCPI	6.35
:SOURce2:SWEep[:FREQuency]:STEP:LOGarithmic	0.01 to 100PCT	not-SCPI	6.35
:STATus:PRESet			6.36
:STATus:QUEue [:NEXT]?			6.36
:SYSTem:COMMunicate:GPIB[:SELF]:ADDRess	1 to 30		6.37
:SYSTem:COMMunicate:SERial:BAUD	1200| 2400| 4800| 9600| 19200| 38400| 57600| 115200		6.38
:SYSTem:COMMunicate:SERial:BITS	7 | 8		6.38
:SYSTem:COMMunicate:SERial:SBITs	1 | 2		6.38
:SYSTem:COMMunicate:SERial:CONTrol:RTS	ON | IBFull | RFR		6.38
:SYSTem:COMMunicate:SERial:PACE	XON | NONE		6.38
:SYSTem:COMMunicate:SERial:PARity	ODD | EVEN | NONE		6.38
:SYSTem:ERRor?			6.38
:SYSTem:PRESet			6.39
:SYSTem:PROTect[1|2|3|4][:STATe]	ON | OFF, Password	not-SCPI	6.39
:SYSTem:SECurity[:STATe]	ON | OFF		6.39
:SYSTem:SERRor?		not-SCPI	6.39
:SYSTem:VERSion?			6.39
:TEST:DIRect	Address, subaddress, hex data string		6.40
:TEST:ASSy	Module, subaddress, Hex data string		6.40
:TEST:RAM?			6.40
:TEST:ROM?			6.40
:TEST:BATTery?			6.40
:TRIGger1|2[:SWEep][:IMMediate]		not-SCPI	6.41
:TRIGger1|2[:SWEep]:SOURce	AUTO | SINGle | EXTernal	not-SCPI	6.42
:TRIGger:PULSe:SOURce	AUTO | EXTernal | EGATed	not-SCPI	6.42
:TRIGger:PULSe:SLOPe	POSitive | NEGative	not-SCPI	6.42

7 Remote Control - Programming Examples

The examples explain the programming of the instrument and can serve as a basis to solve more complex programming tasks.

QuickBASIC has been used as programming language. However, the programs can be translated into other languages.

Including IEC-Bus Library for QuickBasic

```
REM ------ Include IEC-bus library for quickbasic -------
'$INCLUDE: 'c:\qbasic\qbdecl4.bas'
```

Initialization and Default Status

The IEC bus as well as the settings of the instrument are brought into a defined default status at the beginning of every program. Subroutines "`InitController`" and "`InitDevice`" are used to this effect.

Initiate Controller

```
REM ------------ Initiate Instrument -----------
REM InitController
iecaddress% = 28                      'IEC-bus address of the instrument
CALL IBFIND("DEV1", generator%)       'Open port to the instrument
CALL IBPAD(generator%, iecaddress%)   'Inform controller on instrument address
CALL IBTMO(generator%, 11)            'Response time to 1 sec
REM ***************************************************************
```

Initiate Instrument

The IEC-bus status registers and instrument settings of the SML are brought into the default status.

```
REM ------------ Initiate Instrument --------------
REM InitDevice
CALL IBWRT(generator%, "*CLS")          'Reset status register
CALL IBWRT(generator%, "*RST")          'Reset instrument
CALL IBWRT(generator%, "OUTPUT ON")     'Switch on RF output
REM***************************************************************
```

Transmission of Instrument Setting Commands

Output frequency, output level and AM modulation are set in this example. By analogy to the step width setting of the rotary knob, the step width is additionally set for the alteration of the RF frequency in the case of UP and DOWN.

```
REM -------- Instrument setting commands -------------
CALL IBWRT(generator%, "FREQUENCY 250E6")  'RF Frequency 250 MHz
CALL IBWRT(generator%, "POWER -10")        'Output power -10 dBm
CALL IBWRT(generator%, "AM 80")            'AM with modulaton index of 80%
CALL IBWRT(generator%, "AM:INTERNAL:FREQUENCY 3KHZ")
                                           'Modulation frequency 3kHz
CALL IBWRT(generator%, "AM:SOURCE INT")    'Modulation source LF generator
CALL IBWRT(generator%, "FREQUENCY:STEP 12000")
                                           'Step width RF frequency 12 kHz
REM ************************************************************************
```

Switchover to Manual Control

```
REM -------- Switch instrument over to manual control -------
CALL IBLOC(generator%)                'Set instrument to Local state
REM ************************************************************************
```

Reading out Instrument Settings

The settings made in the example above are read out here. The abbreviated commands are used.

```
REM --------- Reading out instrument settings -------------
RFfrequency$ = SPACE$(20)          'Provide text variables with 20 characters
CALL IBWRT(generator%, "FREQ?")    'Request frequency setting
CALL IBRD(generator%, RFfrequency$) 'Read value

RFlevel$ = SPACE$(20)              'Provide text variables with 20 characters
CALL IBWRT(generator%, "POW?")     'Request level setting
CALL IBRD(generator%, RFlevel$)    'Read value

AMmodulationdepth$ = SPACE$(20)    'Provide text variables with 20 characters
CALL IBWRT(generator%, "AM?")      'Request setting of modulation depth
CALL IBRD(generator%, AMmodulationdepth$)'Read value

AMfrequency$ = SPACE$(20)          'Provide text variables with 20 characters
CALL IBWRT(generator%, "AM:INT:FREQ?") 'Request setting of modulation frequency
CALL IBRD(generator%, AMfrequency$) 'Read value

Stepwidth$ = SPACE$(20)            'Provide text variables with 20 characters
CALL IBWRT(generator%, "FREQ:STEP?") 'Request step width setting
CALL IBRD(generator%, Stepwidth $)  'Read value

REM -------- Display values on the screen -------------
PRINT "RF frequency:          "; RFfrequency$,
PRINT "RF level:              "; RFlevel$,
PRINT "AM modulationdepth:    "; AMmodulationdepth$,
PRINT "AM frequency:          "; AMfrequenz$,
PRINT "Step width:            "; stepwidth$
REM************************************************************************
```

Command synchronization

The possibilities for synchronization implemented in the following example are described in Chapter 5, Section "Command Order and Command Synchronization".

```
REM -------- Examples of command synchronization ---------
REM Command ROSCILLATOR:SOURCE INT has a relatively long execution time
REM (over 300ms). It is to be ensured that the next command is only executed
REM when the reference oscillator has settled.

REM -------- First possibility: Use of *WAI ------------
CALL IBWRT(generator%, "ROSCILLATOR:SOURCE INT; *WAI; :FREQUENCY 100MHZ")

REM -------- Second possibility: Use of *OPC? --------
OpcOk$ = SPACE$(2)                  'Space for *OPC? - Provide response
CALL IBWRT(generator%, "ROSCILLATOR:SOURCE INT; *OPC?")
REM -------- here the controller can service other instruments -----------
CALL IBRD(generator%, OpcOk$)        'Wait for "1" from *OPC?

REM -------- Third possibility: Use of *OPC
REM In order to be able to use the service request function in conjugation
REM with a National Instruments GPIB driver, the setting "Disable Auto
REM Serial Poll" must be changed to "yes" by means of IBCONF.

CALL IBWRT(generator%, "*SRE 32")    'Permit service request for ESR
CALL IBWRT(generator%, "*ESE 1")     'Set event-enable bit for
                                     'operation-complete bit
ON PEN GOSUB OpcReady         'Initialization of the service request routine
PEN ON
CALL IBWRT(generator%, "ROSCILLATOR:SOURCE INT; *OPC")
REM Continue main program here.
STOP                                'End of program

OpcReady:
REM As soon as the reference oscillator has settled, this subroutine is
REM  activated
REM  Program suitable reaction to the OPC service request.
ON PEN GOSUB OpcReady               'Enable SRQ routine again
RETURN
REM ************************************************************************
```

Service Request

The service request routine requires an extended initialization of the instrument in which the respective bits of the transition and enable registers are set.
In order to be able to use the service request function in conjugation with a National Instruments GPIB driver, the setting "Disable Auto Serial Poll" must be changed to "yes" by means of IBCONF.

```
REM ---- Example of initialization of the SRQ in the case of errors ------
CALL IBWRT(generatot%, "*CLS")       ’Reset status reporting system
CALL IBWRT(generator%,"*SRE 168")    ’Permit service request for STAT:OPER-,
                                     ’STAT:QUES- and ESR register
CALL IBWRT(generator%,"*ESE 60")     ’Set event-enable bit for command, exe-
                                     ’cution, device-dependent and query error
ON PEN GOSUB Srq                              ’Initialization of the service
                                              ’request routine
PEN ON
REM Continue main program here
STOP                                          ’End of program
```

A service request is then processed in the service request routine.

Note: The variables userN% and userM% must be pre-assigned usefully.

```
Srq:
REM ------------ Service request routine -----------
DO
   SRQFOUND% = 0
   FOR I% = userN% TO userM%                  ’Poll all bus users
        ON ERROR GOTO nouser                  ’No user existing
        CALL IBRSP(I%, STB%)                  ’Serial poll, read status byte
        IF STB% > 0 THEN                      ’This instrument has bits set
                                              ’in the STB
          SRQFOUND% = 1
          IF (STB% AND 16)  > 0 THEN GOSUB Outputqueue
          IF (STB% AND 4)   > 0 THEN GOSUB Failure
          IF (STB% AND 32)  > 0 THEN GOSUB Esrread
        END IF
nouser:
   NEXT I%
LOOP UNTIL SRQFOUND% = 0
ON ERROR GOTO error handling
ON PEN GOSUB Srq: RETURN                      ’Enable SRQ routine again;
                                              ’End of SRQ routine
```

Reading out the status event registers, the output buffer and the error/event queue is effected in subroutines.

```
REM -------- Subroutines for the individual STB bits ------
Outputqueue:                                    'Reading the output buffer
Message$ = SPACE$(100)                          'Make space for response
CALL IBRD(generator%, Message$)
PRINT " Message in output buffer :"; Message$
RETURN

Failure:                                        'Read error queue
ERROR$ = SPACE$(100)                            'Make space for error variable
CALL IBWRT(generator%, "SYSTEM:ERROR?")
CALL IBRD(generator%, ERROR$)
PRINT "Error text :"; ERROR$
RETURN



Esrread:                            'Read Event status register
Esr$ = SPACE$(20)                   'Preallocate blanks to text variable
CALL IBWRT(generator%, "*ESR?")     'Read ESR
CALL IBRD(generator%, Esr$)
IF (VAL(Esr$) AND 1) > 0 THEN PRINT "Operation complete"
IF (VAL(Esr$) AND 4) > 0 THEN GOTO Failure
IF (VAL(Esr$) AND 8) > 0 THEN PRINT "Device dependent error"
IF (VAL(Esr$) AND 16) > 0 THEN GOTO Failure
IF (VAL(Esr$) AND 32) > 0 THEN GOTO Failure
IF (VAL(Esr$) AND 64) > 0 THEN PRINT "User request"
IF (VAL(Esr$) AND 128) > 0 THEN PRINT "Power on"
RETURN
REM ************************************************************************

REM ------------- Error routine ----------------
Error handling:
PRINT "ERROR"                                     'Output error message
STOP                                              ' Stop software
```